

Offene Hypertextsysteme¹

Das Konstanzer Hypertextsystem (KHS) im wissenschaftlichen und technischen Kontext

Anwendungen des Konstanzer Hypertextsystems

Rainer Hammöhner

1. Einleitung

Als eine besondere Eigenschaft offener Hypertextsysteme wurde ihre Flexibilität hinsichtlich ihrer Anwendung hervorgehoben. Der Begriff der Anwendung ist im Zusammenhang mit Hypertext aber nicht unproblematisch. Wir werden deshalb im folgenden also zunächst klären, was wir unter einer Hypertextanwendung verstehen wollen. Ein Szenarium wird dann zur Verdeutlichung des Begriffs beitragen. Aus dem Szenario werden dann Beispielapplikationen abgeleitet, deren Beschreibung den Hauptteil dieser Ausführungen ausmacht.

1.1 Was sind Hypertextanwendungen?

Das Verfassen bzw. der Aufbau von Hypertexten ist eine Tätigkeit, die innerhalb zweier sehr unterschiedlicher Traditionen begriffen werden kann, nämlich der literarischen Tradition der Buchkultur und der erheblich jüngeren technischen der Konstruktion von Informationssystemen. Die jeweils unterschiedliche Ausrichtung schlägt sich auch im Vokabular nieder. Wird im einen Fall von Schreiben oder Verfassen von Hypertexten gesprochen (z.B. [Landow 92, S. 70-100] oder [Nielsen 93, S. 171-181]), so stehen im anderen Fall die Begriffe 'Entwicklung', 'Entwurf' und 'Konstruktion' im Vordergrund (z.B. [Gloor 90] oder [Hofmann & Simon 95]).

Entscheidender als die Begrifflichkeiten sind jedoch Unterschiede in Gestaltungszielen und -methoden. Der Verfasser eines Buches kann auf einen Kanon tradierter Ausdrucksmittel zurückgreifen:

- Normen regeln die physische Gestalt eines Buches. Hier hat der Autor normalerweise wenig Einflußmöglichkeiten.
- Die inhaltliche Gliederung ist weitgehend dem Autor überlassen. Dennoch haben sich auch hier in Form von Textsorten etc. Konventionen ausgebildet, die einen Zusammenhang zwischen Kommunikationsziel und inhaltlicher Aufbereitung herstellen.
- Vermittelnd zwischen physischer und inhaltlicher Struktur wirken paratextuelle Ausdrucksmittel, wie Kapitel- und Paragrapheinteilungen.

Der Autor eines Hypertexts kann auf solche Normen nicht zurückgreifen. Sie sind hier erst im Entstehen begriffen. Insofern ist der Versuch einer Systematisierung von Hypertextstruktur-

¹ Dieser Text ist Teil einer 1997 beim UVK erschienenen Monographie mit dem Titel *Offene Hypertextsysteme. Das Konstanzer Hypertextsystem (KHS) im wissenschaftlichen und technischen Kontext.* (S. 301-375)



Dieser Text ist unter der folgenden Creative Commons Lizenz lizenziert: Attribution-NonCommercial-NoDerivs 2.0 Germany (<http://creativecommons.org/licenses/by-nc-nd/2.0/de/>).

und -visualisierungselementen, wie er im Teil II dieser Arbeit unternommen wurde, als ein Beitrag zur Ausbildung derartiger Konventionen zu verstehen.

Der Rückgriff auf die Begrifflichkeiten der Buchtradition darf jedoch nicht den Blick dafür verstellen, daß auch in den Printmedien schon Gattungen vertreten sind, auf die Begriffe wie 'Schreiben' und 'Autor' nur bedingt anwendbar sind. Exemplarisch seien hier Loseblattsammlungen und Zeitschriften genannt. Hier dominiert eine Makrostruktur, die für alle Exemplare eines Periodikums unveränderlich bleibt, die Gestaltungsfreiheit des Autors, die im Extremfall, wie z.B. im Programmteil einer Fernsehzeitschrift, völlig entzweien kann. Ein Fernsehprogramm ist das Print-Analogon zum Informationssystem, das in einem festgefügten Schema Daten zugänglich macht.

Das Spektrum zwischen frei gestalteten Texten einerseits und rigide strukturierten Texten, die dem Autor keinerlei Freiheit mehr lassen, andererseits muß nun auch vom Medium Hypertext abgedeckt und partiell neu gestaltet werden. Betrachtet man die Entwicklung fortgeschrittenen Hypertextsysteme, so ist hier ein Trend in Richtung auf eine zunehmende Formalisierung der Hypertextstruktur festzustellen. Während der Wert von Hypertext [Nelson 92] zunächst in der freien Assoziation gesehen wurde, wurden im Zusammenhang mit der Entwicklung von anwendungsspezifischen Hypertextsystemen, die auf argumentative Texte zugeschnitten waren (z.B. gIBIS, SEPIA), und zusätzlich motiviert durch das Desorientierungsproblem immer striktere auf Objekttypen beruhende Konfigurationsregeln in Hypertextsysteme eingeführt. Dabei wird von den konkreten medialen Objekten abstrahiert und auf inhaltliche Gesetzmäßigkeiten rekuriert. Soll ein Hypertextsystem, wie z.B. das KHS, nicht auf einen Gegenstandsbereich beschränkt bleiben, so muß eine Vielzahl von Objekttypen verfügbar gemacht werden. Eine Anwendung kann dann als eine Menge von Objekttypen verstanden werden, die minimal erforderlich ist, um unter Berücksichtigung der Konfigurationsregeln einen geschlossenen Hypertext zu erstellen. Das schließt nicht aus, daß ein Hypertext mehrere Anwendungen enthalten kann.

1.2 Koexistenz unterschiedlicher Formalisierungsgrade in Hypertexten

Im vorangegangenen Abschnitt wurden applikationsspezifische Objekttypen und Konfigurationsregeln als ein Weg zur Strukturierung von Hypertexten vorgestellt. Der Preis einer garantierten Konsistenz strikt getypter Hypertexte ist die Gefahr der Überstrukturierung oder auch der verfrühten Strukturierung. Erstere bezieht sich auf das Phänomen, daß komplexe Repräsentationsstrukturen selbst dann benötigt werden, wenn ein simpler Sachverhalt ohne jeglichen Formalisierungsanspruch notiert werden soll. In einem solchen Fall besteht die Gefahr, daß der Autor sich einen einfachen Weg sucht, den intendierten Zweck zu erreichen, der aber nicht unbedingt mit dem vom Designer antizipierten übereinstimmen muß. Vorzeitige Strukturierung liegt dann vor, wenn der Autor gezwungen wird, einen durchaus formalisierungsfähigen Gedanken, der aber die erforderliche intellektuelle Durchdringung noch nicht erlangt hat, auf hohem formalem Niveau auszudrücken. Dies wird in vielen Fällen zur Wahl einer inadäquaten Struktur führen und damit die weitere Ausarbeitung der Ideen eher hindern denn fördern.

Dieses Problem wird von [Haake et al. 94] aufgegriffen und diskutiert. Die Autoren gehen davon aus, daß das zentrale Problem nicht in der Vorgabe eines zu hohen Strukturierungsniveaus liegt, sondern darin, daß der Formalisierungsgrad vom System vorgegeben wird, um den Informationsbedarf systeminterner Algorithmen bedienen zu können. Menschen hingegen sind gewöhnt, sehr flexibel mit Dokumenten umzugehen, die Information in unterschiedlichen Medien auf sehr disparamem Formalisierungsniveau enthalten. Ein Entwurf beginnt häufig mit einer lässig hingeworfenen Handskizze und endet irgendwann in einer hochformalen Beschreibung. Im Zusammenhang einer Hypertextanwendung stellt sich nunmehr die Frage nach dem situativ angemessenen

Formalisierungsniveau und nach dem Grad der Systemunterstützung, der aufgrund der inhaltlichen Erschließbarkeit dieser Dokumente ermöglicht wird.

[Haake et al. 94] ermöglichen eine systematische Untersuchung dieses Problemfeldes, indem sie Hypertexte in einem '*design space*' klassifizieren. Dies ist ein 2-dimensionaler Raum, dessen Koordinaten jeweils definieren, inwieweit Objekttypen vom Nutzer (Leser/Autor) und vom System differenziert werden (s. Abb. 1). In der Abbildung werden die vier Extrempositionen besonders hervorgehoben, die in dieser Metrik möglich sind:

1. Weder Nutzer noch System unterscheiden Objekttypen — dies ist z.B. bei Pixelorientierten Zeichenprogrammen der Fall.
2. Der Nutzer unterscheidet keine Objekttypen. Das System allerdings analysiert die Nutzereingabe und differenziert so verschiedene Objekttypen, die im System repräsentiert werden. Beispielhaft sei hier die Konturerkennung erwähnt, die es einem Rechner ermöglicht, geometrische Figuren in Punktgraphiken zu erkennen und zu repräsentieren.
3. Der Nutzer identifiziert Objekttypen, aber das System repräsentiert diese nicht. Viele bildpunktorientierten Zeichenprogramme erlauben die Eingabe von Schrift, für die auch verschiedene Schrifttypen und -großen festgelegt werden können. Das System repräsentiert die eingegebenen Buchstabenfolgen dann aber nur als Bildpunktmatrizen, so daß im folgenden keine schriftspezifischen Editieroperationen mehr möglich sind.
4. Hochdifferenzierte Objekttypen, die sowohl vom Nutzer unterschieden und genutzt als auch vom System repräsentiert und mit spezifischen Algorithmen manipuliert werden, treten z.B. in vielen CAD-Anwendungen auf.

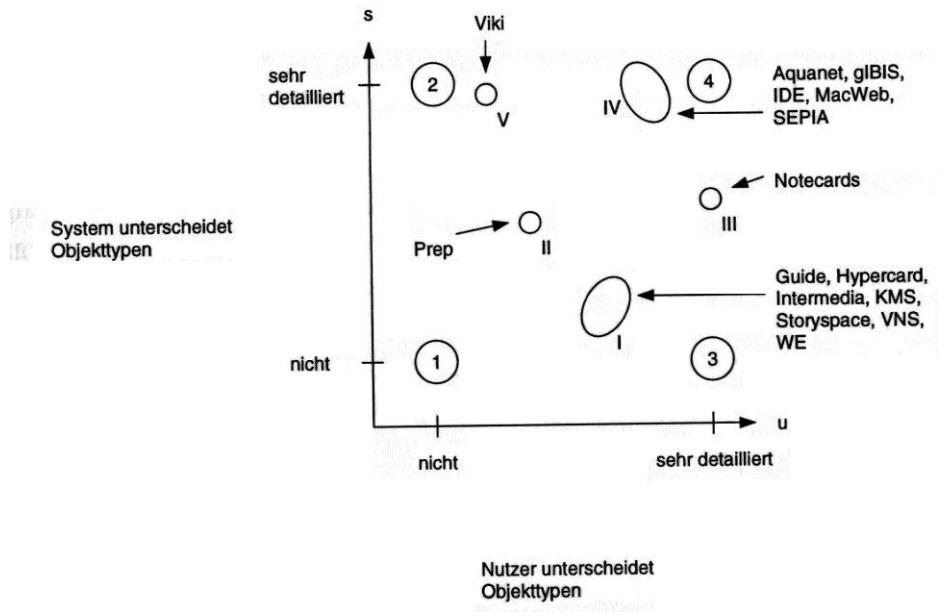


Abbildung 1 Design Space mit existierenden Hypermediasystemen (nach [Haake et al. 94])

[Haake et al. 94] positionieren wichtige existierende Hypertextsysteme wie folgt in diesem Koordinatensystem:

- Eine starke Gruppe bilden die Hypertextsysteme, die intern nur zwischen Knoten und Verknüpfungen unterscheiden. Der Nutzer kann diese Objekte zwar mit Etiketten versehen, die er als typdifferenzierend interpretiert, das System kann jedoch keinen Gebrauch davon machen. Aufgeführt sind Guide [Brown 87], HyperCard, Intermedia [Yankelovich et al. 88], KMS [Akscyn et al. 87], Storyspace [Bolter & Joyce 87], VNS [Shipman et al. 89] und WE [Smith et al. 87].

- Eine zweite umfängliche Gruppe umfaßt Systeme, die Knoten- und Verknüpfungstypen auch intern unterscheiden und nutzen, um Funktionalität bereitzustellen, wie z.B. zur Konsistenzwahrung usw. Das KHS ist zunächst am ehesten in dieser Gruppe einzuordnen. Die weiteren aufgeführten Systeme, wie Aquanet [Marshall et al. 91, Marshall & Rogers 92], gIBIS [Conklin & Begeman 87], IDE [Jordan et al. 89], MacWeb [Nanard & Nanard 91] und SEPIA [Streitz et al. 92], wurden im Zusammenhang mit der Diskussion typisierter Objekte dementsprechend schon wiederholt erwähnt.
- Drei Systeme nehmen individuelle Positionen im Koordinatensystem ein:
 - Die Position von NoteCards [Halasz 88] ergibt sich daraus, daß das System Knotentypen intern repräsentiert. Knotentypen beziehen sich hier ausschließlich auf den Typ der enthaltenen Daten (z.B. Text, Graphik). Verknüpfungen sind nicht typisiert, sondern etikettiert.
 - Der Prep Editor [Neuwirth et al. 90] ist in der Lage, Verknüpfungstypen aus der Entstehungssituation (z.B. Annotation) zu inferieren. Er repräsentiert also Verknüpfungstypen, die der Nutzer nicht explizit unterschieden hat.
 - VIKI [Marshall et al. 94] ist ein Hypertextsystem, das auf räumlicher Anordnung der Knoten basiert (s.a. [Marshall & Shipman 93]). Die Besonderheit dieses Systems ist, daß es aus einer nutzerdefinierten räumlichen Anordnung von Knoten und deren Typen eine inhaltlich sinnvolle Struktur des Hypertexts (Aggregatobjekte, Verknüpfungen) automatisch ableiten kann.

[Haake et al. 94] argumentieren nun, daß ein flexibles, offenes Hypertextsystem möglichst Ausdrucksmittel für alle unterschiedlichen Formalisierungsgrade zur Verfügung stellen soll, so daß der Nutzer entscheiden kann, ob er seine Ideen informell in einer Freihandskizze oder formal in einem Formular ablegen will. Um eine durchgängige Verwendbarkeit von Dokumenten garantieren zu können, ist es dann allerdings erforderlich, Konvertierungsmöglichkeiten zwischen Dokumenten unterschiedlicher Formalisierung bereitzustellen. Mit Dolphin wurde am IPSI ein System geschaffen, das eine *derartige* Funktionalität zur Unterstützung der *Planung und Durchführung von Sitzungen* anbietet, wobei sowohl persönliche Anwesenheit als auch elektronische Kommunikation während der Sitzung vorgesehen sind.

Bevor wir die Transformationsprozesse besprechen, wollen wir kurz darauf eingehen, welche Stufen der Formalisierung vom KHS angeboten und unterstützt werden.

1. Das KHS erlaubt, Texte oder Bilder ohne Hinweis auf ihre Funktion in bestimmten Arbeitsbereichen anzusammeln. Sie können völlig unverknüpft bleiben oder durch frei assoziative Verknüpfungen verbunden werden.
2. Das KHS verfügt über verschiedene Verfahren, Struktur, die dem Nutzer nicht bewußt sein muß, in den Hypertext einzuführen:
 - a. Verfahren der automatischen Indexierung erlauben eine inhaltliche Unterscheidung zwischen Hypertextknoten und definieren eine Zugriffsstruktur für den gesamten Hypertext.
 - b. Bestimmte Knotentypen (Knotentypen, die bestimmte Formen von Terminen beschreiben, Annotation etc.) werden vom System aufgrund der vom Nutzer angestoßenen Aktionen oder anhand der Knoteninhalte ohne weiteres Zutun des Nutzers erkannt und neu erzeugten Knoten zugewiesen.

- c. Automatische Strukturierungsverfahren bewirken den Aufbau von Hypertextstrukturen, die der Nutzer nicht explizit planen muß (z.B. Clustering oder alphabetische Verzeichnisse).
3. Dem Nutzer stehen mehrere Möglichkeiten zur Verfügung, Struktur in den Hypertext einzuführen, die vom System nicht interpretiert werden kann.
- a. Orthogonal zu dem vom System kontrollierten Typsystem kann der Nutzer jedem Hypertextobjekt (Knoten und Verknüpfungen) ein von ihm frei wählbares Etikett zuordnen, das die Funktion des Objekts innerhalb eines nur vom Nutzer kontrollierten Typsystems widerspiegelt.
 - b. Jedem Objekt können individuelle vom Nutzer benannte Attribute zugeordnet werden. Diesen Attributen können als Werte dann aber nur die elementaren Datentypen (Text, Zahl, Intervall) als Ausprägungen zugewiesen werden.
 - c. Hypertextobjekte können zu freien Aggregaten zusammengestellt werden, die im Hypertext mit Hilfe von Kopieroperationen als Bausteine verwendet werden können.
4. Wie bereits in Kapitel 3 in Teil II ausgeführt wurde, bietet das KHS die Möglichkeit, Objekttypen zu definieren, die über spezifische Konfigurationsregeln, Binnenstruktur und differenzierte Funktionalität verfügen. Auch hier sind noch Nuancen im Formalisierungsgrad zu verzeichnen, z.B. hinsichtlich der Kontrolle der internen Struktur von Objekten:
- a. Formulartypen geben eine feste Attributstruktur vor, die allerdings informell erweitert werden kann (s.o.). Es findet eine Kontrolle hinsichtlich der Datentypen der Attributsausprägung statt.
 - b. Auch Frames geben eine Attributstruktur vor, erlauben aber zusätzlich erweiterte Schlußverfahren, wie z.B. Default-Reasoning (s. Kapitel 2).

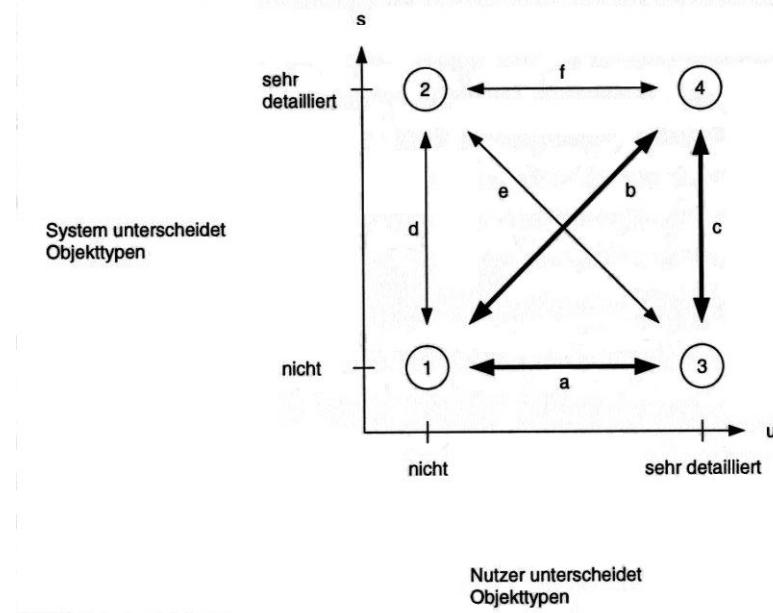


Abbildung 2 Konvertierung im Design Space (nach [Haake et al. 94]). Die fett eingezeichneten Pfeile symbolisieren Transformationen, die von Dolphin bereitgestellt werden.

Im Prinzip sind nun folgende Transformationen zwischen Dokumenten verschiedenen Formalisierungsgrades vorstellbar (s.a. Abb. 2):

- a. Transformation *a* (von 1 nach 3) bewirkt, daß der Nutzer eine vom System nicht unterstützte Formalisierung in eine bisher unstrukturierte Objektmenge einführt:
- o Knotenmengen werden auf mehrere Strukturknoten aufgeteilt.
 - o Information, die unsystematisch in Text oder Graphik eingebettet war, wird in nutzerdefinierten Objektattributen repräsentiert.
 - o Objekte, die der Nutzer mit bestimmten Diskurszielen assoziiert, werden durch nutzervergebene Typbezeichner etikettiert.

Da die Formalisierung nur in der Vorstellung des Nutzers existiert, aber nicht vom System repräsentiert oder gar kontrolliert wird, liegen diese Transformationen vollständig in der Verantwortung des Nutzers, können jedoch vom System unterstützt werden, indem z.B. eine Kontrollliste der vom Nutzer vergebenen Etiketten geführt wird.

Die Umkehroperation besteht im Entfernen von Typbezeichnern, nutzerkontrollierten Attributen bzw. in der Auflösung von Strukturknoten. 1

- b. Die Transformation *b* (von 1 nach 4) setzt nicht oder schwach strukturierte Objekte unmittelbar in systemkontrolliert getypte um, deren Struktur auch vom Nutzer konzeptualisiert wird. Derartige Umformungen treten im KHS auf im Zusammenhang mit der automatischen Analyse schwach strukturierter Daten, wie sie z.B. bei der Integration von e-mail oder den Ergebnissen von Literaturrecherchen auftreten (s.u.). Die Umkehroperation tritt z.B. dann auf, wenn strukturierte Objekte, wie z.B. Formulare, in nicht strukturierten Fließtext umgeformt werden.
- c. Die Transformation *c* (von 3 nach 4) besteht in der Überführung nutzerseitig eingeführter Formalisierungen in solche, die auch vom System kontrolliert werden können. Es gibt zwei Verfahren, wie dies geschehen kann:
- o Nutzerseitig eingeführte Objekttypen werden auf existierende Typen des Systems abgebildet. Nutzerdefinierte Attribute werden soweit möglich auf systemorientierte Attribute abgebildet (die Abbildung erfolgt über den Attributnamen) und andernfalls in ihrem Status belassen.
 - o Es wird ein neuer Systemtyp eingeführt, der alle nutzerdefinierten Attribute in systemkontrollierte umsetzt. Dieser neue Objekttyp kann dann auch mit Konfigurationsregeln versehen werden, welche die bisher im Zusammenhang mit der Nutzung des nutzerdefinierten Typs entwickelten Konventionen formalisieren.

Die Umkehroperation, d.h. die Überführung systemkontrolliert typisierter Objekte in nutzerorientiert typisierte, kann dann von Nutzen sein, wenn basierend auf existierenden Systemtypen eine neue, bisher nur schwach analysierte Applikation aufgebaut werden soll. Die entstehenden Objekte verfügen im Prinzip über die gleiche Struktur, wie von den Systemtypen vorgegeben, haben aber keine spezifische Funktionalität und können leicht modifiziert werden.

- d. Die Transformation *d* (von 1 nach 2) ist relativ ähnlich zur Transformation *b*. Allerdings unterliegen hier die entstehenden Objekttypen allein der Systemkontrolle und müssen nicht vom Nutzer konzeptualisiert werden. Dies ist z.B. bei einer automatischen Indexierung der Fall. Die Umkehroperation führt zu Entfernung der systemkontrollierten Objekte, z.B. des Index.

- e. Die Transformation *e* (von 3 nach 2) unterscheidet sich von der Transformation *c* dadurch, daß die Einführung systemkontrollierter Typen den Nutzer von der Aufgabe, bestimmte Unterscheidungen zu treffen, enthebt. Die Einführung eines automatischen Kassationsverfahrens in hypertextgestützen Archiven z.B. befreit den Benutzer davon, sich um die Archivierungsdauer individueller Dokumente zu kümmern. Er muß nicht mehr beachten, daß jedem Dokument ein formal kontrolliertes Attribut zugeordnet ist, daß die Entstehungszeit betrifft, wie er es im Zusammenhang mit rein intellektuell verwalteten Archiven mußte. Die Umkehroperation, im KHS ohne praktische Relevanz, führt dazu, daß allein systemgesteuerte Strukturierung wieder komplett in die Verantwortung des Nutzers zurückverlagert wird
- f. Die Transformation *f* (von 2 nach 4) macht eine systeminterne Unterscheidung, ein Strukturierungsverfahren explizit für den Nutzer.

Die Umkehroperation sorgt im Sinne des Information Hiding dafür, daß bestimmte systeminterne Unterscheidungen von Objekttypen für den Nutzer transparent werden.

Wie schon hervorgehoben wurde, ist die Unterscheidung und Unterstützung unterschiedlicher Formalisierungsgrade in Hypertextsystemen erforderlich, um dem Nutzer zu gestatten, ein seinen Zielen angemessenes Formalisierungs- und Abstraktionsniveau zu wählen. Im Zusammenhang mit den vorangehend erläuterten Transformationsprozessen ist sie aber auch für die Entwicklung von Hypertextanwendungen von zentraler Bedeutung. Auf diesen Aspekt soll im nächsten Abschnitt näher eingegangen werden.

1.3 Entwickeln von Anwendungen

Die Systematisierung der Integration von Objekten unterschiedlichen Formalisierungsgrades innerhalb eines Hypertexts ermöglicht nun eine vertiefte Analyse der Entwicklung von Hypertextanwendungen. In Abschnitt 1.1 war die Anwendungsentwicklung im Gegensatz zum Schreiben von Hypertexten wie folgt charakterisiert worden:

- o Das Verfassen von Hypertexten betrifft das Erstellen von medialen Objekten (Texten, Graphiken, Formularen) und ihre Organisation innerhalb einer komplexen Hypertextstruktur. Dies geschieht unter Ausnutzung existierender Ausdrucksmittel (Objekttypen, Interaktionsformen etc).
- o Die Entwicklung von Hypertextanwendungen geschieht auf einem höheren Abstraktionsgrad. Hier werden Objekttypen und Interaktionsformen entwickelt, die den Aufbau von Hypertexten zu einem bestimmten Problemfeld durch Bereitstellung normierter Ausdrucksmittel und verbesserte Systemunterstützung erleichtern sollen. Es geht also darum, einen erhöhten Formalisierungsgrad von Hypertexten zu erreichen.

In diesem Zusammenhang erweist sich eine von [Marshall et al. 94] eingeführte Klassifikation von Hypertextsystemen als sehr fruchtbar, die erfaßt, wie Struktur in Hypertexte eingeführt und kontrolliert wird.

- Ein System wird als *permissiv* klassifiziert, wenn keinerlei Typkontrolle vorgenommen wird, wie dies z.B. beim World Wide Web oder Storyspace der Fall ist.
- *Preskriptive* Systeme verfügen über ein festgefügtes Diskursmodell, das durch einen vordefinierten Satz von Objekttypen implementiert wird. Typspezifische Konsistenzregeln werden strikt kontrolliert. Systeme dieses Typs sind z.B. SEPIA und gIBIS.

- *Deskriptive* Systeme, wie z.B. Aquanet, Mac Web oder OVAL [Malone et al. 92], erlauben dem Nutzer, eigene Abstraktionen zu finden und als Objekttypen zu implementieren, die aus abstrakten Typen oder Meta-Schemata abgeleitet werden können.
- Als *emergent* wird ein System dann bezeichnet, wenn aus informellen Strukturen automatisch oder systemunterstützt neue Schemata abgeleitet werden können. Beispielsysteme sind VIKI [Marshall et al. 94] und HOS [Shipman & McCall 94]

Es ist offensichtlich, daß offene Hypertextsysteme in den Bereich der *deskriptiven* oder *emergenten* Systeme fallen müssen. Während *permissive* Systeme zu ausdrucksarm sind, um überhaupt die Unterscheidung von Anwendungsgebieten zu erlauben, sind *preskriptive* Systeme zu strikt auf eine Anwendung hin festgelegt.

1.3.1 Definieren von Objekttypen

Das KHS ist in erster Linie in die Klasse der deskriptiven System einzuordnen. Das System verfügt über eine Menge vordefinierter abstrakter Objekttypen und zugehörige Interaktions- und Präsentationsformen, wie sie im Teil II dieser Arbeit eingeführt wurden. Jeder dieser abstrakten Objekttypen ist durch eine Anzahl von Attributen definiert, die für konkrete Subtypen mit Ausprägungen versehen werden müssen. Einfache Applikationen, die über eigene Konfigurationsregeln verfügen und neue Formulartypen einführen, sind leicht durch Zuweisen von Attributausprägungen aus den abstrakten Objekttypen abzuleiten. Die Definition derartiger Objekttypen ist effizient durch graphische Benutzerschnittstellen mit vordefinierten Auswahlmenüs und Masken zu unterstützen. Sollen sich diese Objekttypen allerdings durch ein individuelles Verhalten auszeichnen, so ist ein gewisser Programmieraufwand einzuplanen. Die Entwicklung von Hypertextapplikationen folgt in diesem Fall weitgehend den Gesetzmäßigkeiten eines konventionellen Softwareentwurfs. Ein derartig kontrollierter Ansatz zur Hypertextentwicklung ist sicherlich dann angemessen, wenn ein gut vorstrukturiertes Problem einer hypertext-orientierten Lösung zugeführt werden soll. Dies ist insbesondere dann gegeben, wenn schon bekannte und vielleicht schon in konventionellen Applikationsprogrammen evaluierte Strukturierungsprinzipien und Algorithmen nur noch in eine Hypertextumgebung integriert werden müssen, um eine erhöhte Datenintegration zu erzielen. Die auftretenden Gegenstände und ihre Beziehungen sind dann in der Regel schon bekannt, so daß primär folgende Fragestellungen zu beantworten sind:

- Welche Mehrwerte verspricht eine Konversion in eine Hypertextapplikation?
- Welche Gegenstände treten auf und wie sind sie auf Objekttypen des Hypertextsystems abzubilden? Wo müssen aus abstrakten Typen neue konkrete zur Unterstützung dieser Applikation abgeleitet werden?
- Wie sind die Objekte der neuen Applikation zu aggregieren? Welche Konfigurationsregeln oder Templates sind zu definieren?

Eine besondere Problematik entsteht bei der Definition von Objekttypen in verteilten Systemen hinsichtlich der Bereitstellung der Funktionalität in den dem Nutzer zugänglichen Client-Programmen. Hier ist zwischen der datenbank- und der datei-basierten Fassung des KHS zu unterscheiden. Erstere steht unter zentraler Kontrolle. Die Typdefinitionen der applikationsspezifischen Hypertextobjekte sind in der objektorientierten Datenbank enthalten. Allein applikationsorientierte Spezialisierungen von Visualisierungs- oder Interaktionsobjekten sind dem Client-Programm gesondert bekannt zu machen. Hier ist ein Protokoll zum Austausch von Software denkbar, das — analog den Paketen von Java — auf einer Zusammenfassung aller einer Applikation zugehörigen Klassen- und Methodendefinitionen zu einer Datei beruht. Jeder Client weiß um die Applikationen, die in ihm installiert sind, und um ihren jeweiligen Version. Wird eine Applikation neu gestartet, so

wird mit dem Server ein Abgleich der Software-Version vorgenommen. Für dateibasierte Hypertexte stellt sich das Problem insofern als komplexer dar, als nicht von einer zentral verwalteten Hypertextbasis auszugehen ist, sondern von einer Vielzahl kleinerer von individuellen Nutzern erstellter Hypertexte. Hier muß jeder Hypertext Referenzen zu den von individuellen Nutzern erstellten Applikationspaketen aufweisen können. Probleme können hier aus folgenden Fragestellungen entstehen:

- Wie können eindeutige Bezeichner für Klassennamen erzwungen werden, obwohl die zugehörigen Applikationen von Nutzern verfaßt wurden, die von ihren Projekten gegenseitig nicht informiert waren?
- Wie kann überprüft werden, ob eine von einem fremden Nutzer erstellte KHS-Applikation unerwünschte Eingriffe in eigene Datenbestände oder Software vornimmt?

Während Namenskonflikte durch die Zuordnung von paketspezifischen Namensräumen vermieden werden können, sind unerlaubte Eingriffe in Datenbestände oder Software in Smalltalk nur schwer feststellbar. Hier sind die Konzepte von Java von Bedeutung, einer Programmiersprache, die für die Nutzung von Software konzipiert wurde, die durch elektronische Netze verfügbar gemacht wird.

1.3.2 Emergenz von Hypertextstrukturen

Die Arbeitsgrundlage *deskriptiver* Hypertextsysteme ist, daß sich die Struktur einer Applikation *a priori* festlegen läßt, bevor noch das erste Dokument, das diesem Anwendungsgebiet zuzuordnen ist, erstellt wäre. Diese Prämisse trifft jedoch nur in wenigen Situationen zu, wenn etwa, wie oben schon erwähnt, bewährte Anwendungen auf Hypertext übertragen werden. Ansonsten ist diese Vorgehensweise, die auf eine sehr frühzeitige Festlegung von Struktur hinarbeitet, eher inadäquat. Hypertext ist ein sehr junges Medium, das seine innere Logik und daraus folgende Strukturierungsprinzipien erst noch entwickeln muß. In diesem Sinne scheint ein evolutionärer Ansatz angemessener, der es erlaubt, Hypertextstrukturen im Gebrauch zu definieren und weiterzuentwickeln. Diese Vorgehensweise sei am Beispiel der Ableitung von Struktur aus räumlicher Anordnung veranschaulicht, wie sie idealtypisch im System VIKI [Marshall et al. 94] ermöglicht wird. Die Basishypothese, die dem System zugrunde liegt, ist die, daß Menschen gewöhnt sind, Dinge räumlich anzurichten, um sie inhaltlich zu gruppieren. Das System erlaubt also, Hypertextobjekte in einer graphischen Präsentation in eine räumliche Beziehung zu setzen. Analyseprozesse ermitteln dann zusammengehörige Gruppen und leiten daraus Objekttypen ab. Die folgende Sequenz von Abbildungen Abb. 3—6, analog zu [Marshall & Shipman 93]) veranschaulicht diesen Prozeß.

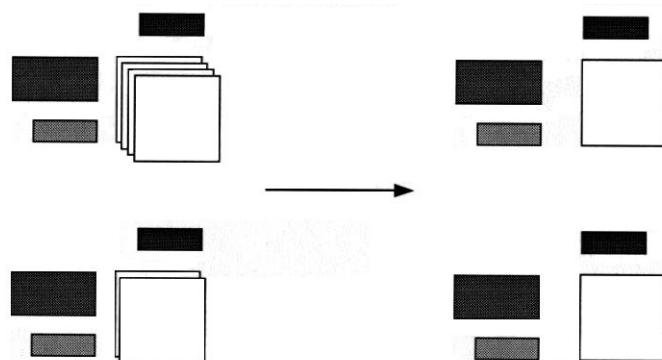


Abbildung 3 Emergenz von Struktur aus räumlicher Anordnung 1: Überlappende einheitlich strukturierte Objekte werden zusammengefaßt, ihr Typ bleibt erhalten. Aus der Art der Anordnung kann zusätzlich noch erschlossen werden, ob es sich um ein allgemeines Aggregat oder eine geordnete Liste handelt.

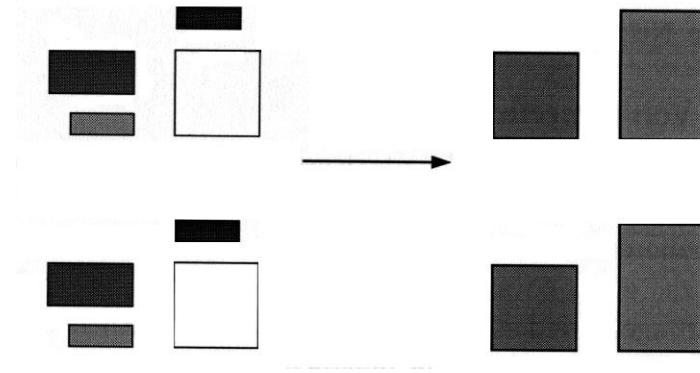


Abbildung 4 Emergenz von Struktur aus räumlicher Anordnung 2: Vier unterschiedliche Aggregate werden aufgefunden, von denen jeweils zwei gleichen Typs sind.

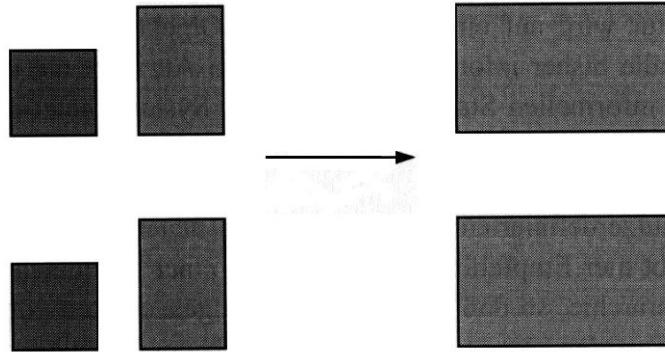


Abbildung 5 Emergenz von Struktur aus räumlicher Anordnung 3: Eine Aggregattyp einer höheren Ebene wird identifiziert.

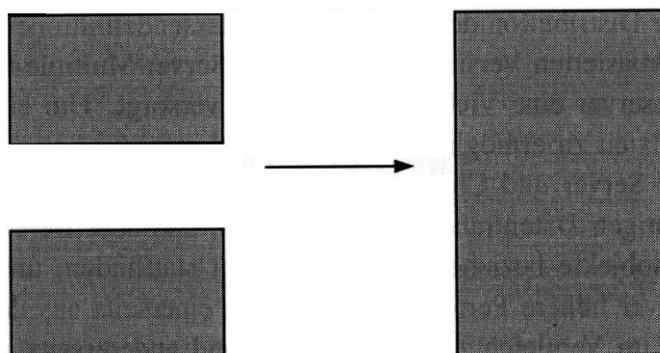


Abbildung 6 Emergenz von Struktur aus räumlicher Anordnung 4: Zwei Aggregate gleichen Typs werden zusammengefaßt, der Typ bleibt erhalten.

Das KHS ist, wie oben schon erwähnt, eher auf deskriptive Typdefinition angelegt. Dennoch ermöglicht auch das KHS Vorgehensweisen, die auf eine evolutionäre Entwicklung von Typen abzielen. Diese wurden schon in der Debatte um den Formalisierungsgrad von Hypertextinhalten angedeutet. War dort das Augenmerk eher auf das synchrone Existieren von Objekten unterschiedlicher Formalisierungstiefe gerichtet, steht hier die diachrone Entwicklung von formalen Objekten aus informellen im Fokus.

Selbst auf der elementarsten Ebene sind im KHS folgende Objekttypen zu unterscheiden: Medienknoten, Strukturknoten, Verknüpfungen. Die Evolution einer Anwendung kann dann ablaufen wie folgt:

- Zunächst sind nur Textknoten gegeben, die in einen Strukturknoten ohne weitere Konfigurationsregeln eingebettet sind.

- Die Textknoten werden auf verschiedene Strukturknoten verteilt. Dieser Prozeß kann durch Verfahren des statistischen Clustering unterstützt werden.
- Eine Innenstruktur der Knoten, in Form nutzerspezifizierter Formulare, wird ausdifferenziert. Das System bietet bei diesem Formalisierungsschritt insofern Unterstützung, als Textknoten, die bestimmten typographischen Konventionen gehorchen, automatisch in Formulare umgesetzt werden können.
- Für Knoten einheitlicher Struktur werden individuelle Typetiketten vergeben.
- Die nutzerdefinierte Struktur wird in eine systemkontrollierte überführt (entsprechend der Transformation c in der Abb. 2), Hier bestehen zwei Möglichkeiten:
 - Die Struktur wird auf einen existierenden Objekttyp abgebildet, dieser deckt entweder alle bisher informell beschriebenen Attribute ab, oder ein Teil bleibt in seinem informellen Status bestehen. Das System bietet entsprechende Typkonversionen an und empfiehlt jeweils die am besten übereinstimmenden Systemtypen.
 - Aus der nutzerdefinierten Struktur wird ein neuer Systemtyp abgeleitet. Das System gibt hier Empfehlungen hinsichtlich einer adäquaten Einordnung in die Klassenhierarchie, so daß möglichst viele Eigenschaften von einem abstrakten Typ geerbt werden können. Ein etwaig vergebenes Typetikett wird als Vorschlag für den Bezeichner der neuen Klasse übernommen, insofern nicht schon ein derartiger Klassename existiert.

Ein zentrales Problem der Überführung nutzerdefinierter Objekttypen in systemkontrollierte besteht in der Distribution der entstehenden Klassendefinitionen. KHS, zumindest in seiner datenbankbasierten Version, ist ein Single-Server/Multiple-Client-System, d.h. daß ein Datenbankserver eine Vielzahl von Clients versorgt. Um eine effiziente Interaktion mit Hypertexten zu ermöglichen, wurde die Ausführung von Berechnungs- und Suchprozessen auf Server und dienten so verteilt, daß Suchprozesse auf dem Server ablaufen, um unnötigen Datentransfer zu vermeiden, daß aber Berechnungen, die sich auf einzelne Datenobjekte beziehen, auf dem Client stattfinden, um die im Vergleich zum Datenbankserver höhere Performance, die sich einerseits aus der größeren Effizienz von Smalltalk im Vergleich zu Smalltalk-DB und andererseits aus der vermutlich *geringeren Auslastung des Clients ergibt, auszunutzen*. Dazu muß der Client aber über alle Klassendefinitionen der auftretenden Hypertextobjekte verfügen. Wird also von einem Nutzer ein neuer Objekttyp angelegt, so muß er allen anderen Clients verfügbar gemacht werden. Weiterhin entsteht das Problem, welchen Nutzern ein so weitgehender Eingriff in die Software-Struktur des Hypertextsystems gestattet werden soll. Im KHS wird das Problem bisher wie folgt gelöst:

- Die Definition eigener Objekttypen wird nur bevorrechtigten Nutzern gestattet, die Distribution der zugehörigen Software wird durch die allgemeinen Routinen im Rahmen der Weiterentwicklung des Systems geregelt.
- Für den nicht bevorrechtigten Nutzer ergibt sich eine alternative Variante, strukturierte Datentypen zu definieren. Im Zusammenhang mit der Diskussion von Wissensrepräsentation als Hypertextanwendung (s. Kapitel 2) werden Frames als Objekttypen eingeführt, die es erlauben, innerhalb eines Hypertexts Metainformation einzuführen, welche Struktur und Verhalten anderer Hypertextobjekte determiniert. Diese Typdefinitionen sind dann zwar auf einen einzelnen Hypertext beschränkt, führen dafür aber nicht zu der oben beschriebenen Distributionsproblematik.

1.4 Der Wissenschaftlerarbeitsplatz

Der Auswahl der größeren Anwendungen, die bisher für das KHS realisiert wurden und in den nächsten Kapiteln beschrieben werden, liegen widerstrebende Interessen zugrunde. Es soll gezeigt werden, daß relativ breit gestreute Anwendungsbereiche mit dem hier vorgestellten Ansatz zu behandeln sind. Gleichzeitig muß aber auch zur Geltung kommen, daß durch die Integration von Daten verschiedener Applikationen und ihre gegenseitige Bezugnahme Mehrwerte entstehen. Da das KHS in einer relativ kleinen Projektgruppe entstand, durften die Beispiele auch nicht zu umfanglich geraten. Bis übergreifendes Anwendungsbereich wurde die Unterstützung eines Arbeitsplatzes für Wissenschaftler gewählt. Der Ehrgeiz des Projektteams wurde so durch den Eigennutz befriedigt, immerhin sind einige der hier beschriebenen Anwendungen mittlerweile in der FG Informationswissenschaft der Universität Konstanz im täglichen Einsatz.

Wenngleich die Auswahl der Anwendungen, wie oben schon angedeutet, nicht zuletzt von Erwägungen der Projektpräsentation und von Eigenbedarf geleitet war, fügen sich diese durchaus in ein übergreifendes Szenarium ein.

Die tägliche Arbeit eines Wissenschaftlers ist weitestgehend durch den Umgang mit Dokumenten bestimmt:

- Zentrale Tätigkeit ist das Verfassen wissenschaftlicher Publikationen. In das Umfeld der hypertextorientierten Autorensysteme wurde bereits soviel Forschungsaufwand investiert, in Deutschland vor allem im Umfeld des SEPIA-Projekts, daß wir dieses Gebiet, auch des großen Aufwands wegen, nicht bearbeitet haben.
- Mit der Tätigkeit des wissenschaftlichen Schreibens untrennbar verbunden sind die Literaturrecherche und die Verwaltung von Bibliographien und Literaturverzeichnissen (s. Abschnitt 3.5).
- Kommunikation mit Fachkollegen erfolgt in zunehmendem Maße elektronisch via e-mail. Dabei werden nicht nur kurze Mitteilungen, sondern auch ganze Dokumente ausgetauscht, so daß e-mail intensiv mit anderen Anwendungen verknüpft ist (s. Abschnitt 3.2).
- Auch der Wissenschaftler benötigt die üblichen Büroanwendungen, wie Adress- und Terminverwaltung (s. Abschnitte 3.3 und 3.2.2.3). Hervorzuheben ist hier weniger die innere Komplexität dieser Anwendungen als ihre Integration zu der in sich geschlossenen Funktionalität eines Büroarbeitsplatzes.
- Jede wissenschaftliche Tätigkeit bezieht sich auf einen komplexen Gegenstand, der zur Protokollierung oder Publikation von Forschungsergebnissen besondere Dokumenttypen, wie z.B. Versuchsprotokolle, verlangt. Als Forschungsgegenstand wurde hier das Software-Engineering mit seiner Vielzahl von Dokumenttypen ausgewählt (s. Kapitel 4). Diese Auswahl motiviert sich in erster Linie daraus, daß am Beispiel des Software-Engineering ganz eigene, von üblichen Dokumenten abweichende Strukturprinzipien aufgezeigt werden können. Natürlich kann dieses umfassende Gebiet im Rahmen dieser Arbeit nur tangiert werden.
- An den Universitäten ist Forschung immer auch mit Lehre verknüpft. Hier ermöglichten offene Hypertextsysteme neue Formen selbstbestimmten Lernens [Garbe 95] und die gemeinsame Verwaltung und Nutzung von Lehrmaterialien [Volst 94],

Eine Sonderstellung im Kanon der Anwendungen nimmt die Wissensrepräsentation ein. Wissensrepräsentation ist hier nur als Anwendung des Hypertextmodells des KHS zu betrachten, d.h. daß konkrete Objekttypen abgeleitet wurden, die eine formale Repräsentation von Wissen ermöglichen. Innerhalb des oben beschriebenen Anwendungsszenarios

hingegen ist Wissensrepräsentation nicht als eine weitere Anwendung zu verstehen, sondern als eine methodische Grundlage, die manche Anwendungsfunktionen - z.B. die regelbasierte Klassifikation von Hypertextknoten überhaupt erst ermöglicht. Diese wiederum ist die Grundlage für die Erstellung und Verwaltung nicht-linärer, dynamischer Dossiers, welche die heterogenen oben erwähnten Informationsressourcen aufgabenspezifisch zusammenführen [Kuhlen 94].

2. Wissensrepräsentation

Das Verhältnis von KI, insbesondere bezogen auf die Thematik Wissensrepräsentation und Expertensysteme, und Hypertext wird von der mit Hypertext befaßten Forschergemeinde ambivalent gesehen. Ein Impuls für die rasante Entwicklung von Hypertext Ende der 80er Jahre lässt sich auch aus dem Versagen der Forschung im Bereich der KI ableiten, Verfahren auch für den Durchschnittsanwender bereitzustellen, die ihm helfen, die ständig anwachsende Informationsflut zu bewältigen (s.a. [Nielsen 93, S. 14]). Hypertext stellt der "Rechnerintelligenz" die natürliche Intelligenz entgegen, deren Entfaltung durch adäquate Aufbereitung der Information gefördert wird.

Trotz oder vielleicht auch wegen dieses Konkurrenzverhältnisses fanden schon recht frühzeitig Methoden und Begrifflichkeiten aus der Wissensrepräsentation ihren Weg in die Hypertextforschung. Insbesondere die Analogie zwischen Hypertext und semantischen Netzen wurde betont [Jonassen 89] und fand in Hypertextsystemen ihren Niederschlag, die in erster Linie als Schnittstellen zu konzeptuellen Netzwerken zu verstehen sind (z.B. [Collier 87]). Entsprechend der Distanzierung von den Ambitionen der KI findet allerdings eine Akzentverschiebung statt von der automatischen Evaluation semantischer Netze aufgrund von Inferenzverfahren, die Teil des Repräsentationsmodells sind, hin zu individuellem Lernen, das Resultat des interaktiv navigierenden Umgangs mit Visualisierungen dieser Netze ist (z.B. [McAleese 90]). So wird nicht nur die Integration von Wissen in die persönlichen kognitiven Strukturen erleichtert, sondern auch dem Autor ein Instrument zur Externalisierung seines Wissens angeboten, das ihm zunächst keine als artifiziell empfundene Linearisierung seiner Assoziationen abverlangt [Streitz 90].

Nicht zuletzt die Diskussion um die "seven issues" [Halasz 88] und die dort erhobene Forderung nach weitergehender Strukturierung der Hypertexte führen zur Einführung stärker systemkontrollierter Strukturen in Hypertexte. Aquanet [Marshall et al. 91] z.B. mit seinem an framebasierte Repräsentationssprachen angelehnten Hypertextmodell wurde mit dem Ziel entwickelt, die Möglichkeiten von Hypertext auch für eine formalisierte Repräsentation von Wissen auszuloten. Das Hypertextmodell von Mac Web [Nanard & Nanard 91] wiederum beruht auf einem semantischen Netz mit getypten Knoten und Kanten, die jeweils mit typspezifischen Methoden, die in einer Skriptsprache definiert werden, versehen werden können. In diesem Zusammenhang unterscheiden [Nanard & Nanard 91] zwischen Information als für den Nutzer interessanten Daten und Wissen als strukturierten und damit auch vom Computer auswertbaren Daten. Wenngleich die begriffliche Unterscheidung uns eher unglücklich erscheint, so ist die Betonung des Zusammenspiels von Struktur und Inferenzverfahren im Zusammenhang mit einem Hypertextsystem von Bedeutung. Der Ansatz von [Kuhlen 91, S. 59 ff.] erscheint hier als zielführender, der als Wissen *den Bestand an (gesichterten, begründbaren) Modellen über Objekte bzw. Objektbereiche und Sachverhalte versteht*, die in Büchern oder auch Hypertexten zur Darstellung kommen, d.h. repräsentiert werden können. Es ist also zwischen Repräsentationen von Wissen zu unterscheiden, die aufgrund ihrer formalen Aufbereitung bereits algorithmisch verarbeitbar sind und solchen, die es bisher nicht sind. Wichtig ist jedoch, daß [Nanard & Nanard 91] die Koexistenz formaler, automatisch auswertbarer und informeller Strukturen in einem Hypertext für sinnvoll halten.

[Rüssel 90] hebt in diesem Zusammenhang hervor, daß unter Einsatz unterschiedlicher Inferenztechniken die gleiche Struktur, d.h. der gleiche Hypertext, aus verschiedenen Perspektiven gesehen und mit verschiedenen Ergebnissen ausgewertet werden kann. Das Verhältnis von Hypertext und Wissensrepräsentation ist auch deshalb relativ komplex, weil es unter sehr verschiedenen Gesichtspunkten betrachten werden kann [Tochtermann & Zink 95]:

- **Hypertext als Wissensrepräsentationssprache:** Dies entspricht in etwa der Perspektive, die in dem voranstehenden kurzen historischen Abriß eingenommen wurde. Hypertext wird als Alternative zu den Wissensrepräsentationssprachen der KI angesehen. Dabei wird die Adaption von erprobten Modellen aus der KI nicht ausgeschlossen, sondern eher begrüßt. Vielmehr werden mittlerweile Systeme, welche das mediale Potential von Hypermediasystemen und die Inferenzmöglichkeiten von Expertensystemen in sich vereinen, als eine neue Klasse von Systemen gesehen, für die sich die Bezeichnung *Expertextsystem* [Rada 91] einzubürgern beginnt.
- **Hypertext als Gegenstand wissensbasierter Verfahren:** Nach relativ kurzer Euphorie wurden schnell die Schattenseiten der Netzwerkstruktur von Hypertext erkannt, die Gefahr der Desorientierung und des *Informational Overload*. Neben anderen Verfahren, wie Typisierung und Bereitstellung von Übersichten, wurden auch Ansätze zur wissensbasierten Unterstützung von Hypertextnavigation und -Präsentation vorgeschlagen. [Weyer & Bornig 85] sehen wissensbasierte Verfahren zur semantischen Filterung von Daten einer elektronischen Enzyklopädie vor, in Abschnitt 4.4 in Teil II dieser Arbeit wurden Systeme vorgestellt, die Wissen über den Nutzer und über den Diskursbereich zur Auswahl der situativ relevanten Information nutzen. Das jeweils angestrebte Ziel determiniert hier auch das Repräsentationsverfahren:
 - Begriffliches Wissen, das zum Aufbau semantischer Verknüpfungen, der automatischen Indexierung oder der Suchfrageerweiterung dient, wird bevorzugt durch semantische Netze mit *Spreading Activation* [Clitherow et al. 89], [Collier 87] oder Frames [Koh & Chua 89].
 - Regelwissen zur Strukturierung von Hypertexten [Knopik & Ryser 89] oder zur Auswertung von Nutzermodellen, z.B. [Nicolson 89].
- **Hypertext zur Unterstützung wissensbasierter Systeme:** Die speziellen Strukturierungs- und Interaktionsverfahren von Hypertextsystemen lassen sie geeignet erscheinen, bestimmte Defizite wissensbasierter Systeme auszugleichen:
 - Hypertextsysteme sind besonders geeignet, Erklärungskomponenten von Expertensystemen zu realisieren. Tritt eine problematische Situation auf, so verweist das Expertensystem auf eine entsprechende Hypertextseite, die eine verbale Erklärung der entsprechenden Regel gibt. Von hier ausgehend kann der Nutzer dann weitere Zusammenhänge explorativ erkunden [Yetim 92, Yetim 94]. Ein Beispiel für derartige Systeme ist Janus [Fischer & McCall 89], das Konstruktionsregeln und Argumentationen, die diese Regeln untermauern, in einem hybriden Ansatz vereint, der ein Hypertextsystem (Document Examiner) mit einer Regelkomponente kombiniert.
 - Die Koexistenz verschiedener Formalisierungsgrade lässt Hypertext als ein adäquates Instrument zur Wissensakquisition von Expertensystemen erscheinen. Der Experte kann sein Wissen informell als Text zur Verfügung stellen. Von dieser Ausgangsbasis kann es über mehrere Formalisierungsschritte in Regel-form transformiert werden (s. z.B. [Snaprud & Kaindl 92]).

Die jeweilige Sichtweise auf das Verhältnis von Hypertext und Wissensrepräsentation bedeutet auch eine unterschiedliche Gewichtung der Integration von formalem und in-

formellem Wissen. Werden Regeln zur Unterstützung der Navigation im Hypertext genutzt, so besteht keine Notwendigkeit, diese Regeln als Teil des Hypertext zu betrachten. Ushir [Nicolson & Thomlinson 91], z.B., ist ein Hypertextsystem, das Navigationsregeln separat vom Hypertext als Prolog-Klauseln repräsentiert, während [Evans 90] ein Expertensystem vorstellt, dessen Regeln auf spezialisierten HyperCard-Karten repräsentiert sind.

Die Integration von wissensbasierten Verfahren in das KHS sollte nicht durch eine der oben erläuterten Perspektiven vorbestimmt sein. Die Offenheit des Systems und der Anspruch, verschiedene Grade an Formalisierung abdecken zu können, legen allerdings unabhängig vom Einsatzziel der Wissensrepräsentationssprache einen Ansatz nahe, der eine hohe gegenseitige Durchdringung ermöglicht. Ein derartiges Modell zur Integration formaler Wissensrepräsentation und informeller medialer Objekte innerhalb eines Hypertexts stellen [Kaindl & Snaprud 91] vor, das darauf beruht, daß der gesamte Hypertext in zwei Halbräume aufgeteilt wird. Jedes Medienobjekt wird in jeweils einem Halbraum durch eine formale, framebasierte und in dem anderen durch eine informelle, meist textuelle, mitunter aber auch bildliche Repräsentation vertreten. Die beiden Repräsentationen werden durch spezielle Verknüpfungen verbunden. Die Menge der formalen Repräsentationen wiederum ist strukturiert durch Typ-Subtyp-Beziehungen, sowie Prototyp-Instanz-Beziehung zwischen Frames [Snaprud & Kaindl 94]. Dieser Ansatz ist vielversprechend, wenngleich es als problematisch erscheint, strikte Dichotomien zwischen formaler und informaler/semitformaler Repräsentation aufzubauen, wobei der formale Teil dem Rechner zur Auswertung vorbehalten ist und der informale grundsätzlich eine für den Nutzer gedachte erläuternde Funktion hat. Hier ist eine Flexibilisierung wünschenswert. Die Auswahl des Frame-Formalismus allerdings hat sich für die Verwendung von Hypertext zur Wissensrepräsentation als paradigmatisch erwiesen. Dies dürfte in erster Linie darauf zurückzuführen sein, daß strukturierte Formen der Wissensrepräsentation am besten mit den Strukturprinzipien von Hypertext harmonieren. Auf die Parallelen zu semantischen Netzen wurde ja bereits hingewiesen. Im Vergleich zu semantischen Netzen bieten Frames den Vorteil, daß die Aggregation von Daten zu 'Chunks', die mit Hypertexteinheiten identifiziert werden können, erleichtert wird. Dementsprechend wird begriffliches und faktisches Wissen auch innerhalb des KHS durch Frames modelliert. Zur Darstellung regelhafter Zusammenhänge werden zusätzlich Produktionsregeln verwendet, die sich gut mit objektzentrierten Modellierungsverfahren zu einem hybriden Repräsentationssystem kombinieren lassen [Puppe 91]. Die in das KHS integrierten Wissensrepräsentationsformate werden wie folgt eingeführt: Nach einer kurzen Darstellung des Formalismus wird beschrieben, wie die jeweiligen Strukturen auf das Hypertextmodell abgebildet werden. Für umfassenderen Informationsbedarf sei hier auf die Übersichtsliteratur (z.B. [Frost 86, Reimer 91, Puppe 93]), insbesondere aber auf [Zink 95] verwiesen, der eine umfassendere Darstellung der Integration von KHS und Wissensrepräsentation gibt.

2.1 Frames

Der Begriff des 'Frame' leitet sich aus dem Schemabegriff der Gestaltpsychologie ab und damit der Vorstellung, daß Objekte oder Situationen durch eine Menge von Merkmalen und Merkmalsausprägungen beschrieben werden können, daß aber bei Fehlen von exaktem Wissen über die Merkmalsausprägungen sogenannte 'Defaults' zum Tragen kommen und in ihrem Zusammenhang einen Stereotyp bilden, der dennoch als Grundlage für weitere Schlußfolgerungen dienen kann. Aus dieser Grundidee hat sich mittlerweile eine Vielzahl von Repräsentationssprachen entwickelt, die aber in ihren Kernkonstrukten weitgehend übereinstimmen.

2.1.1 Basisstrukturen

Die in das KHS integrierte Wissensrepräsentationssprache kann als eine einfache, restriktierte Form einer KL-ONE-Sprache [Puppe 91, S. 47] (vgl. z.B. [Brachman & Schmolze 85])

aufgefaßt werden. Konzepte werden durch Frames modelliert, deren Slots die Konzeptmerkmale darstellen. Generische Konzepte werden durch sogenannte Metaframes modelliert, deren Attribute durch eine Kollektion von Metaslots beschrieben werden. Aus ihnen können Instanzen, sogenannte Datenframes, mit konkreten Merkmalsausprägungen (Datenslots) abgeleitet werden. Jedem Datenslot entspricht dabei genau ein Metaslot, mit den jeweiligen Attributen (s.u.).

Zusätzlich zu Konzepten, die in der Wissensbasis durch Frames definiert sind, kann auch auf sogenannte atomare Konzepte rekurriert werden, die durch die Datentypen der Programmierumgebung bereitgestellt werden. Im Fall des KHS sind dies vor allem die Klassen, die das Hypertextmodell implementieren. Slots verfügen über folgende Eintragsbedingungen:

- Anhand des **Objekttyps** kann festgelegt werden, welchem Typ (atomar oder definiert) die einzutragenden Objekte angehören müssen.
- Die **Kardinalität** definiert, wieviel Ausprägungen die jeweils durch den Slot definierte Eigenschaft haben kann (Anzahl der zugehörigen Datenslots). Dabei sind Angaben zur **Minimalzahl**, **Maximalzahl** möglich. Ein **Standardwert** gibt vor, wieviele Datenslots bei der Erzeugung einer Instanz automatisch angelegt werden.

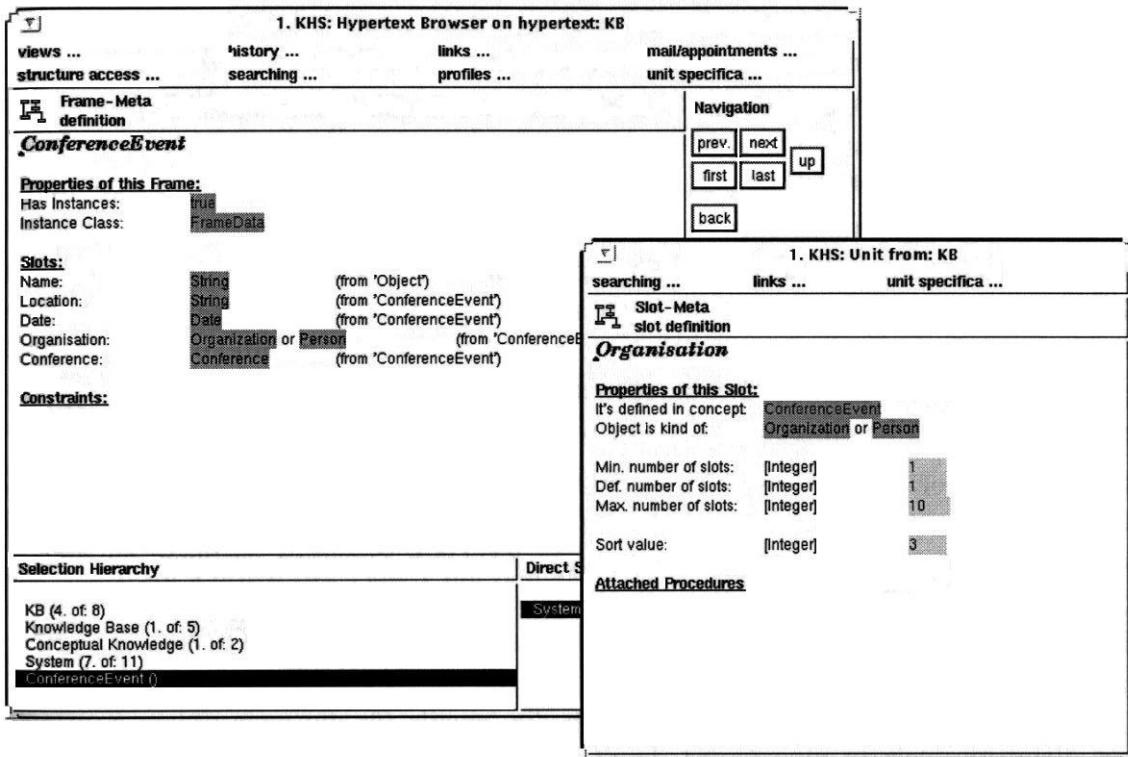


Abbildung 7 Darstellung eines Metaframe in einem Hypertextbrowser mit zusätzlich eingeblendet er Slotdefinition.

Weiterhin kann prozedural definiertes Wissen über 'attached procedures' mit Slots verbunden werden.

- **default** gibt den Default-Wert eines Slots zurück,
- **ifNeeded** wird aufgerufen, wenn auf den Inhalt eines leeren Datenslots zugegriffen wird. Diese Prozedur stellt ein Verfahren bereit, dann aus redundanten Daten die benötigte Information zu berechnen.

- **ifChanged** wird nach Ändern des Inhalts eines Datenslots aufgerufen, z.B. um redundant repräsentierte Daten anzugeleichen,
- **ifAdded** wird nach Hinzufügen eines Datenslot aktiviert.

2.1.2 Beziehungen zwischen Frames

Frames können auf verschiedenerlei Arten miteinander in Beziehung gesetzt werden, die auf Struktur und Inhalt beruhen, z.B. Ähnlichkeit, ausschließende Attributmengen, o.ä. [Frost 86, S. 471]. Von zentraler Bedeutung für den Repräsentationsformalismus 'sind jedoch zwei Relationstypen, die Beziehung zwischen generischem Konzept (Metaframe) und Individualkonzept (Datenframe) und die Spezialisierungsrelation zwischen zwei generischen Konzepten. Durch die Instanzbeziehung wird ein konkretes Objekt einem Typ zugeordnet. Die in der Typbeschreibung festgelegten Merkmale sind sowohl in Prototyp und Instanz gegeben, für die Instanz können Merkmalsausprägungen definiert sein, sonst wird auf die Defaults zurückgegriffen. Die Spezialisierungsrelation ordnet einem allgemeineren ein spezielleres Konzept mit kleinerer Extension unter. Die Spezialisierung erfolgt durch Angabe zusätzlicher Merkmale oder durch die Einschränkung eines gemeinsamen Merkmals. Die Merkmale des allgemeineren Konzepts müssen auf jeden Fall auch für das speziellere Konzept anwendbar sein. Die Existenz dieser Struktureigenschaften der jeweiligen Frames und das Bestehen der Spezialisierungs- bzw. Instanzbeziehungen bedingen sich gegenseitig. Sind die Merkmale von Frames bekannt, so kann auf die bestehenden Framebeziehungen geschlossen werden, sind jedoch die Beziehung und der abstraktere Frame bekannt, so kann auf die Merkmale (Slots und ihre Attribute) geschlossen werden. Im letzten Fall spricht man für gewöhnlich von Vererbung². In manchen Repräsentationssystemen können sowohl Spezialisierung als auch Instanziierung nur 1:n-Beziehungen sein, d.h. ein abstrakter Begriff kann über mehrere Spezialisierungen verfügen, aber nicht umgekehrt. Kann ein Konzept, wie es auch im KHS der Fall ist, mehrere Superkonzepte haben, so können hinsichtlich der Vererbung Konflikte auftreten, indem z.B. unterschiedlich attribuierte Slots das gleiche Merkmal repräsentieren. Das KHS erlaubt in diesem Fall nur die Vererbung eines dieser Slots, die Entscheidung über die Auswahl wird zur Zeit dem Nutzer, der eine Wissensbasis aufbaut, überlassen. Es sind aus der Literatur allerdings bereits komplexe Verfahren zur Auflösung derartiger Konflikte bekannt (z.B. für OBJLOG [Dugerdil 98]).

Eine wichtige Anwendung dieser Eigenschaft framebasierter System ist die Mustererkennung, d.h. einerseits, daß die Erfassung von Merkmalen beobachteter Objekte oder Situationen anhand der modellierten Eigenschaften generischer Objekte erfolgt, und andererseits, daß die Beobachtungen dann zu klassifizieren sind, indem konkrete Objekte entweder als alternative Repräsentationen schon bekannter Individuen oder als Ausprägungen generisch beschriebener erkannt werden (matching).

Ist ein auf die gefundene Merkmalskombination passender Frame nicht vorhanden, so muß ein neues generisches Konzept definiert werden und als Spezialisierung eines bereits bekannten eingeführt werden (Subsumption). KL-ONE-Sprachen verfügen zu diesem Zweck über einen Klassifikator (s. z.B. [Brachman & Schmolze 85] oder [Reimer 89]), der anhand eines vorgegebenen Individualkonzepts das generische Konzept findet, unter welches das Individualkonzept zu subsumieren ist. Kriterien für die Klassifikation sind die beobachteten Merkmale, wie auch die jeweils für die generischen Konzepte definierten Beschränkungen für Merkmalsausprägungen. Der Aspekt der Ausnutzung der Wissensbasis als mustererkennendes System wurde allerdings im Rahmen der Forschung am KHS bisher nicht berücksichtigt,

² Eine gute Übersicht über Aspekte der Vererbung vermittelt ein von [Lenzerini et al. 89] herausgegebener Sammelband.

schließt aber für die Weiterentwicklung des Systems ein erhebliches Potential ein. Zu denken ist hier z.B. an die Modellierung von Dialogsituationen durch Skripts, die dann zur Klassifikation des konkreten Navigationsverhaltens genutzt werden können, so daß eine situative Navigationsunterstützung ermöglicht wird. Auf die mögliche Verwendung von Subsumption beim Aufbau von Hypertexten wird in den folgenden Beispielen noch eingegangen werden.

Eine einfachere, weitere Anwendung terminologischen Wissens innerhalb des KHS stellt die Suchfrageerweiterung für thematische Filterung und Suche dar. Hier werden nach einem Ansatz der *Spreading Activation* (s. z.B. [Cohen & Kjeldsen 87]) ausgehend von gegebenen gewichteten Termen Gewichtungen entlang terminologischer Relationen innerhalb der Wissensbasis propagiert. Alle Terme, deren Gewicht einen Schwellwert übersteigt, werden dann in das Suchprofil aufgenommen.

2.1.3 Abbildung von Frames auf Hypertextobjekte

Alle Konstrukte der oben beschriebenen Repräsentationssprache werden als Spezialisierungen von Objekttypen definiert, die zum Kern des Hypertextmodells des KHS gehören. Aggregierende Beziehungen, wie die Frame-Slot und Slot-Eintragsbeziehung, werden auf die Beziehung zwischen Strukturknoten und ihren Teilen abgebildet, d.h. daß Frames und Slots als spezialisierte Strukturknoten aufzufassen sind. Die Kontrolle von Eintragsbedingungen erfolgt dann nach dem gleichen Prinzip, nach dem auch für Strukturknoten ermittelt wird, welche Teilknoten eingetragen werden können.

Als spezialisierte Strukturknoten verfügen Frames und Slots auch über Funktionen zum Einfügen von Teilknoten (Slots, Sloteinträge). Hier ist nur zu beachten, daß die für Strukturknoten definierten generischen Folgeoperationen dahingehend erweitert werden, daß die Konsistenz des Framemodells gewahrt bleibt, indem z.B. Vererbungsbeziehungen verändert werden.

Andere semantische Beziehungen zwischen Frames, wie Spezialisierung und Instanziierung, werden auf Hypertextverknüpfungen abgebildet. Wie auch bei anderen Verknüpfungstypen werden Existenzbedingungen für diese Verknüpfungen festgelegt. Die Vererbung von Eigenschaften zwischen Frames wird dann beim Erzeugen einer Verknüpfungsausprägung angestoßen, d.h. die Vererbung erfolgt im KHS zur Definitionszeit und nicht zur Anfragezeit (vgl. [Reimer 91, S. 88]).

2.2 Produktionsregeln

Im Zusammenhang mit Aufbau und Nutzung von Hypertexten lassen sich häufig regelhafte Zusammenhänge formulieren, wie

- wenn ein Knoten des Typs x mit den Merkmalsausprägungen y, z erzeugt wurde, so ist er in folgende explizit oder regelhaft benannte Strukturknoten einzufügen, oder
- wenn der Nutzer die Knoten x und y gelesen hat, so ist der Knoten z eine sinnvolle Fortsetzung der Lektüre.

Derartige Zusammenhänge lassen sich gut durch sogenannte Produktionsregeln modellieren, die jeweils aus einem Bedingungsteil und einem Aktionsteil bestehen. Der Bedingungsteil bezieht sich dabei auf eine Faktenbasis, hier den Hypertext oder die Dialoghistorie, während der Aktionsteil relativ beliebige Effekte haben kann, in der Regel aber eine Manipulation der Faktenbasis bewirken wird. Zusammen mit einem Interpreter bilden Faktenbasis und eine Menge von Produktionsregeln ein Produktionssystem. Der Verarbeitungsprozeß eines Produktionssystems, das im Modus der Vorwärtsverkettung betrieben wird, umfaßt folgende Schritte:

- Alle Regeln werden ermittelt, deren Bedingungsteil erfüllt ist.
- Aus diesen Regeln wird eine ausgewählt, dabei kommen Meta-Regeln zum Tragen, auf die hier nicht näher eingegangen werden soll (s. z.B. [Reimer 91, S. 56 f]).
- Es wird geprüft, ob ein vorgegebenes Ziel erreicht ist, oder die Menge der ausführbaren Regeln leer ist. In diesem Fall terminiert der Prozeß, andernfalls wird der Verarbeitungszyklus erneut durchlaufen.

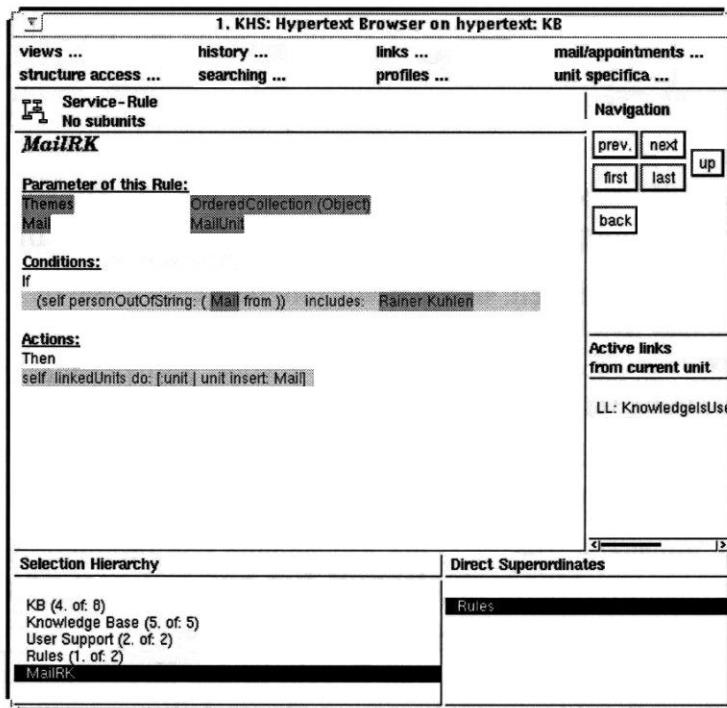


Abbildung 8 Regel zur automatischen Archivierung von e-mail – Wenn als Absender des zu klassifizierenden Briefs ‚Rainer Kuhlen‘ verzeichnet ist, so wird diese in den Folder eingefügt, mit dem diese Regel assoziiert ist.

Während die oben erwähnten Beispiele die Anwendung von Produktionsregeln zur Unterstützung eines Hypertextsystems zumindest intuitiv motivieren — mit Ushir [Nicolson & Thomlinson 91] existiert auch ein System, das Regeln zur Navigationsplanung einsetzt —, so bleibt doch die Integration von Regeln in den Hypertext selbst noch näher zu begründen. Wir wollen hier zwei Argumente hervorheben:

- Im Zusammenhang mit dem Aufbau von persönlichen Archiven ist der Nutzer genötigt, selbst Archivierungsregeln zu benennen. Es ist hier von Vorteil, wenn er nicht auf weitere Werkzeuge als diejenigen des Hypertextsystems zurückgreifen muß.
- Ein Problem im Zusammenhang regelbasierter Systeme ist der Aufwand, der bei der Suche nach ausführbaren Regeln entsteht. Dieser Aufwand lässt sich zum Teil durch eine adäquate Strukturierung der Regelmenge erheblich reduzieren. Dies kann auf eine dem Nutzer leicht nachvollziehbare Art und Weise durch Hypertextstrukturen geschehen:
 - Präferenzrelationen zwischen Regeln können durch Verknüpfungen dargestellt werden. Sie können genutzt werden, wenn die Bedingungsteile mehrerer Regeln erfüllt sind.
 - Strukturknoten können Regelmengen in mehrere separate Regelbasen partitionieren, deren Auswahl wiederum durch Meta-Regeln gesteuert wird.

Das für das KHS realisierte Produktionssystem ist weitgehend von dem System NéOpus beeinflußt, das die Einbettung von regelbasiertem Schließen in eine objektorientierte

Umgebung zum Gegenstand hat. Dementsprechend werden sowohl Bedingungsteil als auch Aktionsteil einer Regel in der Syntax einer objektorientierten Programmiersprache (hier: Smalltalk) definiert. Darüber hinaus können externe Variable in den Körper einer Regel eingeführt werden (s. z.B Abb. 8). Während jedoch NéOpus Regeln unmittelbar als Methoden des objektorientierten Systems interpretiert und damit eine Klassenhierarchie zur Gliederung der Regelmenge nutzt, ist im KHS eine Regel durch einen spezialisierten Hypertextknoten repräsentiert. Die Gliederung der Regelmenge erfolgt durch Strukturknoten.

2.3 Zusammenfassung

Die Darstellung von KHS als wissensrepräsentierendes System an dieser Stelle hat zugegebenermaßen etwas Fragmentarisches. Dies ist in erster Linie darauf zurückzuführen, daß hier ein Eindruck der aktuellen Forschung vermittelt wird, ohne daß den noch ausstehenden Originalpublikationen vorgegriffen werden soll. Der hier gegebene Einblick kann dementsprechend nur die Funktion haben, einerseits das Potential von Hypertext im Umfeld der Wissensrepräsentation anzudeuten und andererseits soviel an Detailinformation zusätzlich bereitzustellen, wie zum Verständnis der im weiteren Text aufgeführten Beispiele erforderlich ist. Insbesondere im Zusammenhang mit der automatischen Archivierung von Hypertexteinheiten (s. Abschnitt 3.1) wird auf die in diesem Kapitel eingeführten Methoden zurückgegriffen.

3. Computerunterstützte Gruppenarbeit

Die Arbeit von Wissenschaftlern ist in hohem Maß geprägt von der kooperativen Arbeit in gemeinsamen Forschungsgegenständen, eine Tätigkeit, die ständige Kommunikation und die gemeinsame Nutzung von Informationsressourcen (z.B. Literatur) und auch eine gemeinsame Terminplanung erfordert. Für alle diese Aufgaben haben sich in den letzten Jahren im Rahmen einer zunehmenden Computerunterstützung von Arbeitsplätzen mehr oder weniger standardisierte Werkzeuge herausgebildet³. Eine effiziente computerorientierte Unterstützung der Gruppenarbeit erfordert jedoch eine Maximum an Integration für diese zunächst separat behandelten Tätigkeiten. [Teufel et al. 95] unterteilen diesen Funktionsraum in die folgenden Bereiche:

- Kommunikation:
 - Elektronische Postsysteme
 - Konferenzsysteme
- Gemeinsame Informationsräume
- Workflow Management
- Workgroup Computing:
 - Planungssysteme
 - Gruppeneditoren
 - Entscheidungs- und Sitzungsunterstützungssysteme

Wir wollen im folgenden einen Rahmen aufspannen, wie ein solcher Funktionsraum auf ein offenes Hypermediasystem wie das KHS abgebildet werden kann.

Dabei werden einzelne Bereiche, wie z.B. e-mail, als relativ detailliert ausgearbeitete Beispiele für die Tragfähigkeit des Ansatzes eingeführt, während andere Bereiche nur

³ Einen Überblick über den Stand der Kunst geben [Davenport & McKim 95].

skizziert werden können. Bereiche, für die kein besonderer Nutzen aus der Hypertexttechnologie erwachsen dürfte, wie z.B. Teleconferencing, werden komplett ausgespart.

Die Annahme, die dem hier vorgestellten Ansatz zu Grunde liegt, ist die, daß Hypertext nur durch eine adäquate Strukturierung des Datenmaterials eine angemessene Unterstützung der Zusammenarbeit erzielen kann (vgl. [Greif & Sarin 87]). Die zentrale Funktion ist also, die Strukturierung und Verwaltung eines Informationsraums (s.a. Abb., der allen anderen Funktionen zur Nutzung offensteht. Struktur und Inhalt dieses gemeinsamen Informationsraums bilden damit das Wissen einer Organisation sowohl in Bezug auf Faktenwissen als auch auf Aspekte der Zuständigkeit und Zusammenarbeit (vgl. [Conklin 93]). Dieses Wissen umfaßt Terminologien, Fakten, Mail, Memos etc., also Daten sehr unterschiedlichen Formalisierungsgrades [Schatz 93], die, wie ja schon hervorgehoben wurde (s. Abschnitt 1.2), von modernen Hypermediasystemen gut repräsentiert werden können.

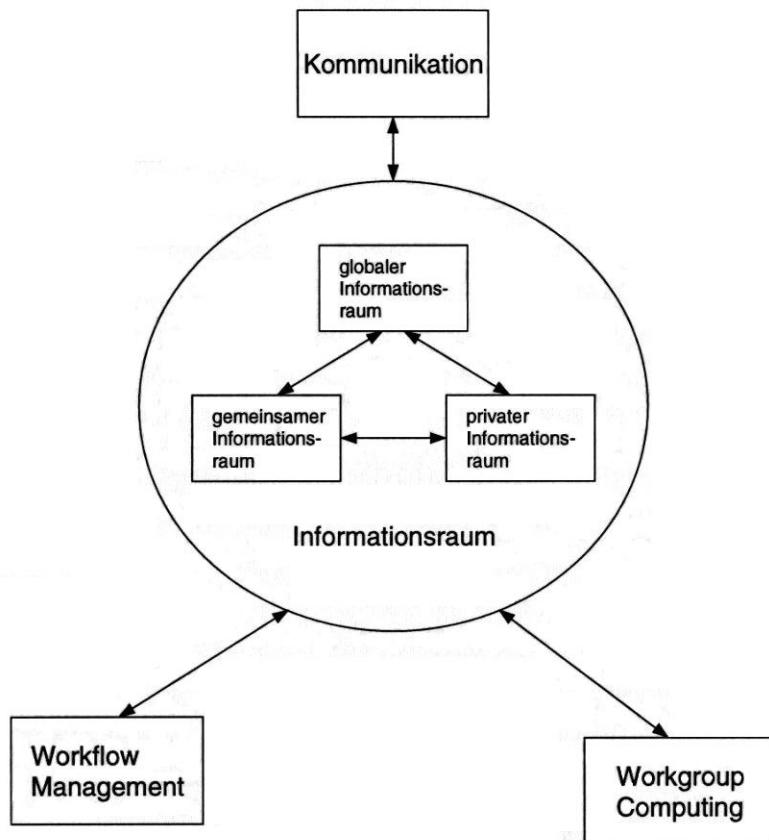


Abbildung 9 Ein strukturierter Informationsraum als steuernde Instanz für Gruppenarbeitsprozesse

3.1 Aufbau und Strukturierung des Informationsraums

Wie bereits bei der allgemeinen Diskussion des Hypertextmodells des KHS hervorgehoben wurde, verfügen elaborierte Hypertextsysteme über zwei Strukturierungsebenen:

- Strukturierung durch Verknüpfungen zielt auf eine relationale Beziehung zwischen einzelnen Hypertextobjekten ab, die Inhalt oder Funktion dieser Objekte betreffen kann. Diese Beziehungen können durchaus applikationsspezifisch sein.
- Strukturierung durch Strukturknoten erreicht eine globale Strukturierung des Hypertexts. Durch hierarchische Dekomposition werden Bereiche des Hypertexts festgelegt, die bestimmten Themen, Nutzern oder Aufgaben vorbehalten bleiben.

Eine globale Strukturierung des Informationsraums durch Strukturknoten ist nach folgenden orthogonalen Kriterien zu konzipieren (s.a. [Richartz & Rüdebusch 90]):

- **Ordnungsstruktur:** Die (Poly-)Hierarchie der Strukturknoten dient dazu, den Nutzer möglichst effizient zu Informationseinheiten zu führen. Dabei ist ein Optimum zu finden zwischen der Tiefe der Hierarchie (Anzahl der Navigationsentscheidungen) und ihrer Breite (Komplexität der Navigationsentscheidungen). Umgekehrt kann diese Ordnungsstruktur auch genutzt werden, um Hypertexteinheiten automatisch in den Hypertext einzufügen. In dem Maße, wie die Ordnungsstrukturen nicht aus intuitiven, sondern algorithmisierbaren Kriterien, die womöglich auf formalisierten Attributen semistrukturerter Knoten beruhen, abgeleitet werden, können sie auch rechnergestützt aufgebaut und erweitert werden.
- **Berechtigungsstruktur:** Die Struktur der Hypertexte sollte so geartet sein, daß der Leser nur Zugriff auf die Teilbereiche des Hypertexts erhält, die er auch aufgrund seiner Tätigkeit oder sonstiger Befugnis lesen darf. Wie bereits in der Einführung des Hypertextmodells des KHS erläutert wurde (s. Abschnitt 2.1 in Teil II), besteht ein enger Zusammenhang zwischen Zugriffsrechten und Strukturhierarchie.
- **Funktionsstruktur:** Die Struktur des Hypertexts kann nicht nur hinsichtlich der Optimierung des Zugriffs auf bestimmte Inhalte und die zugehörigen Zugriffsberechtigungen hin optimiert werden, sondern auch auf den Zweck, der mit bestimmten in einen Kontext eingeordneten Hypertexteinheiten verfolgt werden soll. Beispielsweise seien hier zunächst zwei Funktionsgruppen benannt, die dann in der weiteren Ausführung noch ausführlicher behandelt werden:
 - **Archivierung:** Die Strukturknoten sind daraufhin angelegt, informationelle Einheiten für einen Zeitraum aufzunehmen, um einem bestimmten Klientel effizienten Zugriff auf diese Informationen zu erlauben. Derartige Archive können privat, gruppenspezifisch oder global sein.
 - **Normierte Bearbeitung:** Funktion eines Strukturknotens kann es auch sein, Hypertexteinheiten nur so lange aufzunehmen, bis von jemandem ein bestimmter Vorgang an den Daten durchgeführt wird, worauf diese gezielt weitergeleitet werden.

Diese Ordnungsprinzipien können nunmehr genutzt werden, um eine regelbasierte Einordnung von Hypertextknoten in eine bestehende Struktur zu erreichen. Dies wird erforderlich, wenn ein neuer Knoten ohne Angabe eines Kontexts entweder vom Nutzer oder einem externen Prozeß (z.B. e-mail) erzeugt wird, oder wenn ein Knoten modifiziert wird, so daß eine Neueinordnung erforderlich wird. Regelbasierte Verfahren zum automatischen Aufbau von Archivstrukturen wurden zunächst im Zusammenhang mit Mailsystemen entwickelt — das bekannteste Beispiel dürfte die *Information Lens* sein [Malone et al. 86, Malone et al. 87a, Malone et al. 87b], vgl. jedoch auch das System MAFIA [Lutz et al. 90] oder [Pollock 88] —, wurden jedoch auf beliebige semistrukturierte Objekte ausgeweitet. Aus der *Information Lens* wurde die *ObjectLens* entwickelt [Lai et al. 88]. Klassifikationsprozesse werden durch sogenannte Agenten abgewickelt, die spezifisch für den Informationstyp (z.B. *Mail* oder *Termin*) sind. Jedem Agenten ist eine Menge von Regeln zugeordnet, die jeweils in ihrem Bedingungsteil aus einem Beispieltemplate bestehen, das mit dem zu klassifizierenden Objekt verglichen wird, und einer Aktion, die bei Übereinstimmung ausgeführt wird. Die Aktion besteht zumeist darin, daß das Datenelement in einen bestimmten Ordner eingestellt wird. Trifft also ein neuer Brief ein, so wird der Agent zur Mailklassifikation aktiv, vergleicht das eingetroffene Objekt mit allen Beispielformularen seiner Regeln und führt die Aktionsteile der zutreffenden Regeln aus. Der Vorteil dieses Verfahrens liegt darin, daß nur so viele Vergleichsoperationen auszuführen sind, wie Regeln vorhanden sind. Ein

signifikanter Nachteil besteht aber darin, daß Veränderungen der Ordnerstruktur grundsätzlich auch zu Reformulierung der Regeln führen müssen. Eine Beziehung zwischen Regelstruktur und Ordnerstruktur ist nicht ersichtlich.

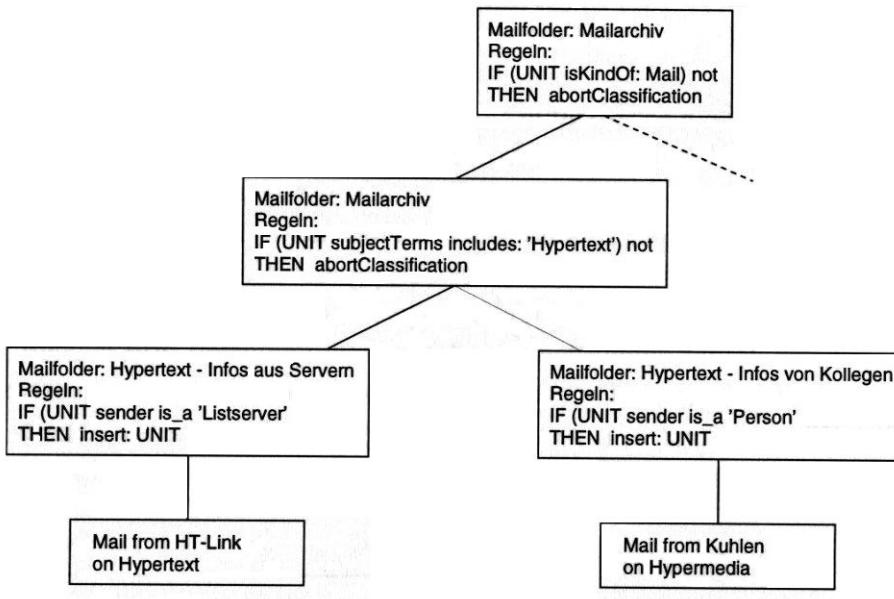


Abbildung 10 Regelbasierte Klassifikation — Ziel ist hier, Mail zum Thema Hypertext, Hypermedia danach zu unterscheiden, ob sie aus einem Listserver stammt, oder von einer Person. Dieses Beispiel ist inhaltsgleich zu dem in Abb. 12, welches das gleiche Ziel durch Frame-Matching erreicht. Die in diesem Beispiel genutzten terminologischen Beziehungen sind aus Abb. 12 zu entnehmen.

3.1.1 Regelbasierte Klassifikation im KHS

Das KHS verfolgt hier einen komplexeren Ansatz, der einen erhöhten Berechnungsaufwand mit einer verbesserten Strukturierungsmöglichkeit belohnt. Einem Agenten wird hier kein Regelsatz zugeordnet, sondern nur ein bestimmter Verknüpfungstyp und ein Startknoten, der als Strukturknoten die Wurzel der zu berücksichtigenden Ordnungsstruktur bildet (s.a. Abb. 10). Ist dann ein Knoten zu klassifizieren, so werden die folgenden Schritte durchlaufen:

- Ausgehend von einem Strukturknoten, beim ersten Durchlauf ist dies der vorgegebene Startknoten, werden alle Regelobjekte (s. Abschnitt 2.2) zu einer Regelmenge zusammengestellt, die durch den vorgegebenen Verknüpfungstyp erreichbar sind.
- Es werden die Regeln herausgefiltert, deren Bedingungsteil erfüllt ist, wenn der zu klassifizierende Knoten als Parameter eingesetzt wird (s.a. Abb. 8). Die Bedingungen können sich dabei auf Eigenschaften der beteiligten Knoten beziehen, aber auch auf das im Hypertext durch Frames repräsentierte terminologische oder organisationsspezifische Wissen oder das Nutzerprofil des Nutzers, der die Klassifikation durchführen lässt.
- Die gefundenen Regeln werden unter Ausnutzung der zwischen den Regeln durch Verknüpfungen etablierten Präferenzrelationen geordnet.
- Die Aktionsteile der Regeln werden in der somit gegebenen Reihenfolge ausgeführt. Dabei sind zunächst folgende Aktionen möglich:
 - Einfügen des Knotens in den betroffenen Strukturknoten.
 - Löschen des Knotens aus dem Strukturknoten. Die Einführung dieses Aktionsstyps erlaubt, Strukturknoten zu definieren, die an Neueintrag und Verbleib eines Knotens unterschiedliche Bedingungen stellen. Wird ein Knoten neu

klassifiziert, so wird nur dann aus den Knoten, in die er schon eingetragen ist, entfernt, wenn die jeweilige Löschregel aktiv wird.

- Fortsetzung des Klassifikationsprozesses mit allen untergeordneten Strukturknoten.
- Fortsetzung der Klassifikation mit allen untergeordneten Strukturknoten, allerdings mit verändertem Klassifikationsverfahren. Dieser Aktionstyp erlaubt, für Teilzweige des Hypertexts spezialisierte Klassifikationsverfahren einzusetzen, ohne den regelbasierten Gesamtrahmen verlassen zu müssen.
- Beenden der Klassifikation für diese Teilhierarchie. Ein Abbruch des Klassifikationsprozesses ist auch dann gegeben, wenn ein Strukturknoten gefunden wird, dem keine Regel zugeordnet ist, deren Bedingungsteil erfüllt wäre. Eine explizite Beendigung des Prozesses wird als Aktion nur deshalb vorgesehen, um eine Wiederholung von Ausschlußbedingungen in weiteren Regeln, die dann allerdings nachgeordnet sein müssen, vermeiden zu können.

3.1.2 Alternative Klassifikationsstrategien

Eine alternative Klassifikationsstrategie, die wie oben beschrieben in die regelbasierte Klassifikation eingebettet werden kann, ist die Klassifikation aufgrund des Übereinstimmungsgrades von Deskriptormengen (vgl. [Hammwöhner & Rittberger 93, Hammwöhner & Kuhlen 94]). Für KHS-Knotentypen sind verschiedene Verfahren der automatischen Indexierung definiert, die dann zum Einsatz kommen, wenn der Inhalt eines Knotens modifiziert wird. Diese können allein auf der Häufigkeit von Textterminen beruhen oder auch die innere Struktur von Knoten berücksichtigen. Dies kann z.B. auch für die Eliminierung von Stoppwörtern relevant sein, indem bei Formularen vorgegeben werden kann, welche Merkmale nicht sinnvoll für eine Indexierung herangezogen werden können. Als Resultat der Indexierung wird dem Knoten also eine Menge von gewichteten Termen zugeordnet. Auch Strukturknoten können verschlagwortet werden. Die Klassifikation von Knoten kann nunmehr so erfolgen, daß sie in die Strukturknoten eingefügt werden, deren Deskriptoren am besten mit den eigenen übereinstimmen (alle, deren Ähnlichkeit einen bestimmten Schwellwert übersteigt, oder falls das bei keinem der Fall ist, der beste verbleibende). Im KHS wird hierzu das Cosinus-Maß verwendet. Dieses Verfahren zeichnet sich dadurch aus, daß es keine hohen Ansprüche an den Nutzer stellt, da nur eine Zuteilung von Schlagworten erforderlich ist. Vor allem aber können mit diesem Verfahren auch Texte klassifiziert werden, die keine formale Struktur aufweisen (s.a. Abb. 11).

Ein weiteres für das KHS konzipiertes, aber zur Zeit noch nicht eingesetztes Klassifikationsverfahren beruht auf Frame-Matching. Dieses Verfahren erfordert, daß der Inhalt des zu klassifizierenden Knotens erschlossen und als Frame repräsentiert wird. Ebenso werden den Strukturknoten Frame-Repräsentationen zugeordnet. Die Zuordnung erfolgt dann zu den Strukturknoten, deren Repräsentation am besten übereinstimmt. Dieses Verfahren ist dem regelbasierten insofern vergleichbar, als gezielt auf Merkmale von Objekten rekuriert werden kann. Es ist allerdings nicht (oder nicht so einfach) möglich, weitere Kontextinformation einzubeziehen. Demgegenüber hat dieses Abgleichverfahren — wie auch schon der Deskriptorenabgleich — den Vorteil, daß es nicht eine vollständige Erfüllung von Bedingungen verlangt, sondern nach einem "Best-match"-Verfahren auch bei immer nur teilweiser Übereinstimmung irgendein Treffer gefunden werden kann (s.a. Abb. 12).

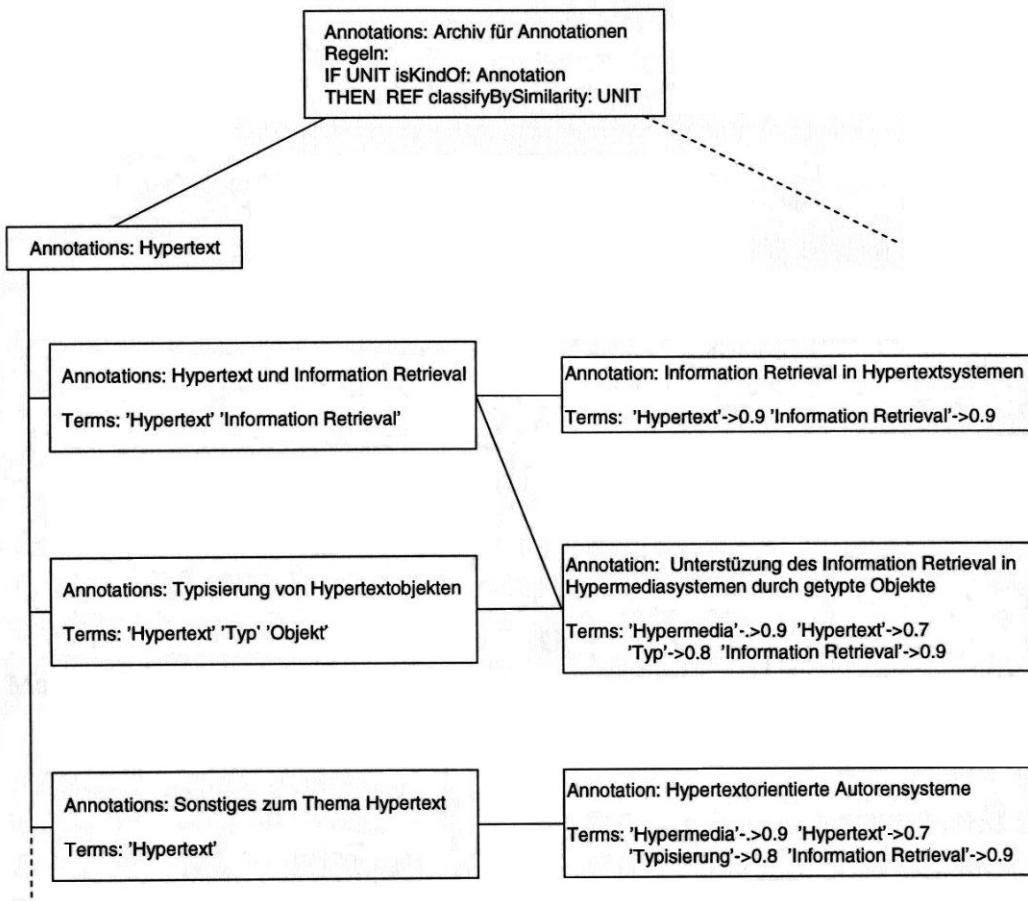


Abbildung 11 Klassifikation anhand der Ähnlichkeit von Deskriptormengen— Die einzelnen Kästen symbolisieren Hypertextknoten. Für jeden Knoten sind zugeordnete Regeln und Deskriptoren angegeben, bei letzteren auch das Gewicht, insofern es kleiner als 1 ist. Die Knoten in der linken Spalte sind Strukturknoten, in die Annotationen, unstrukturierte Textknoten, der rechten Spalte automatisch eingeordnet wurden. Die Konstanten UNIT und REF in der Regel stehen für durch die Regel referenzierten Strukturknoten (REF) und den zu klassifizierenden Knoten (UNIT).

3.1.3 Struktur des Informationsraums

Bisher liegen zu wenig Erfahrungen mit Systemen vor, die über derartig komplex strukturierte Informationsräume verfügen, als daß evaluierte Regeln zur adäquaten, zielorientierten Strukturierung derartiger Hypertexte zu benennen wären. Immerhin sind einige Heuristiken zu formulieren. Information, die von allen genutzt werden soll und darf, steht in größeren Bereichen des Hypertexts zur allgemeinen Verfügung. Die terminalen, informationstragenden Knoten sind dabei jeweils nach verschiedenen Gesichtspunkten geordnet:

- Um eine bessere Wart- und Ergänzbarkeit zu erzielen, sind alle Knoten, die einem Informationstyp zuzuordnen sind, z.B. Literaturreferenzen, Adressen, Termine, jeweils in einem gemeinsamen Teilbaum des Hypertexts angeordnet.
- Teilstrukturen, die einem Informationstyp zugeordnet sind, können typspezifisch mehrere konkurrierende Ordnungsstrukturen aufweisen, wie Ordnung nach Thema, Alphabet, Datum.
- Dossiers stellen Information zu einem Thema zusammen, wobei jegliche Informationstypen auftreten können. Die innere Struktur eines Dossiers kann dann wieder typorientiert sein.

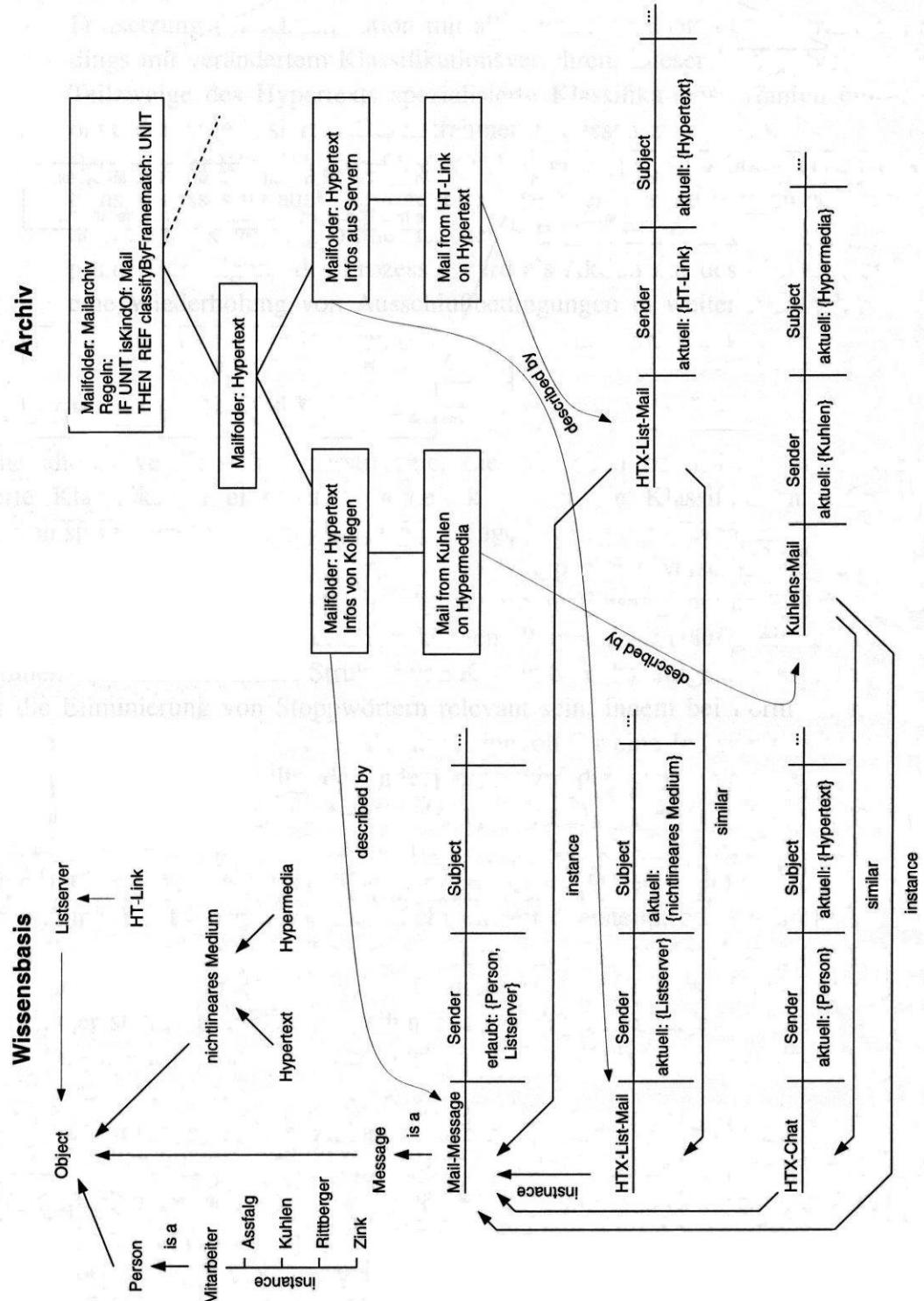


Abbildung 12 Klassifikation aufgrund von Frame-Matching – Frames repräsentieren terminologisches Wissen und Objekttypen (Mail). Frameinstanzen werden den Ordnern als Repräsentationen zugeordnet. Auch die zu klassifizierenden Objekte erhalten Repräsentationen und werden in die Ordner eingefügt, deren Repräsentationen am besten zu der ihres paßt.

Teilbereiche des Hypertexts können auch für die Öffentlichkeit geschlossen werden, indem sie nur bestimmten Gruppen oder Personen zugänglich gemacht werden. Das KHS weist jedem Nutzer einen privaten, nur ihm zugänglichen Arbeitsbereich zu. Gleichermaßen kann für Gruppen geschehen. Diese Mischung von globalen und privaten Informationseinheiten ist nunmehr für die automatische Klassifikation von Hypertextknoten kritisch insofern, als garantiert werden muß, daß vertrauliche Information, die von einem Zugangsberechtigten in

den Hypertext eingebbracht, nicht in einen ungeschützten Ordner eingefügt wird, der in der gleichen Ordnungsstruktur enthalten ist. Dieses Problem soll durch ein Beispiel verdeutlicht werden. Ein Gruppe von Nutzern verwaltet gemeinsam ein Mail-Archiv. Um redundante Datenübertragung zu vermeiden, werden Listserver nur einmal abonniert. Die eintreffenden Nachrichten dieser Server werden dann in einen global zugänglichen Teil des Mailarchivs klassifiziert. Ein Nutzer arbeitet in einem Projekt, in dem vertrauliche Information via e-mail kommuniziert wird, während jemand anderes sich für die gleiche Thematik in einem völlig unverfänglichen Zusammenhang interessiert und diese Information seinen Kollegen offen zur Verfügung stellen möchte. Beide kommunizieren darüber hinaus mit dem gleichen Personenkreis, so daß Regeln, welche Briefe in den vertraulichen und welche in den offenen Ordner einzufügen sind, nur schwer formulierbar sind und Sicherheitslücken leicht entstehen können. Das KHS sieht für diese Problematik mehrere Maßnahmen vor:

- Keinerlei Konflikte können dann auftreten, wenn derjenige, der vertrauliche Information einbinden will, für den Klassifikationsprozeß einen Archivknoten angibt, dessen Teilknoten alle den gleichen Vertraulichkeitsgrad haben. Daraus folgt allerdings auch, daß offene Daten, die in dem zu klassifizierenden Material enthalten sind, nicht global verfügbar gemacht werden können.
- Das KHS sieht, wie bereits in Abschnitt 2.1 in Teil II erläutert, vor, daß Knoten ihre Zugriffsschutzparameter von den Strukturknoten erben, in die sie eingefügt werden. Für sensitive Daten kann dieser Mechanismus so parametriert werden, daß immer der restriktivste Zugriffscode gewählt wird. So wird bewirkt, daß der schutzwürdige Knoten zwar in einen offenen Polder eingefügt wird, aber nur von Berechtigten gelesen werden kann. Die Folge ist allerdings, daß der Eigner dieses Ordners von der Existenz eines solchen Knotens Kenntnis erhält. Obwohl keinerlei Inhaltsinformation vermittelt wird, ist dies doch meist unerwünscht.
- Es besteht die Möglichkeit, den Klassifikationsvorgang so zu gestalten, daß ein Knoten immer nur in die Strukturknoten mit der größten Zugriffssicherheit eingefügt wird. Dies kann bei Regelbasierter Klassifikation mit in die Regeln aufgenommen werden und ist bei den matching-orientierten Klassifikationsverfahren vom System fest vorgesehen.

3.1.4 Kassation

Ebenso wichtig wie ein kontrollierter Aufbau eines Archivs ist die Beseitigung veralteten Materials aus dem Archiv, um ein unkontrolliertes Anwachsen des Bestandes zu vermeiden. Zu diesem Zweck erlaubt das KHS nicht nur die Zuordnung von Klassifikationsregeln, sondern auch die von Kassationsregeln zu Strukturknoten. In regelmäßigen Abständen werden die Bestände anhand dieser Regeln überprüft. Die Bedingungen betreffen dabei im Normalfall die Verweildauer im Archiv, können jedoch auch weitere Aspekte berücksichtigen. So können einzelne Knoten mit Bearbeitungsfristen versehen werden (s. Abschnitt 3.3 zum Thema Termin Verwaltung). Eine Eliminierung dieser Knoten erfolgt dann frühestens, wenn diese Fristen verstrichen sind. Der Aktionsteil der Kassationsregeln bewirkt entweder eine vollautomatische Löschung von obsoleten Knoten oder, wenn es sich um sensitivere Daten handelt, eine Markierung, damit sie nach Überprüfung durch einen Zuständigen aus dem Archiv ausgeschieden werden.

3.2 E-mail

3.2.1 Einführung

Der zur Zeit wahrscheinlich elaborierteste Sektor der computerunterstützten Gruppenarbeit ist die elektronische Post (einen guten Überblick gibt [Palme 95]) als der Spezialfall

asynchroner elektronischer Kommunikation. Der hohe Entwicklungsstand, die Bemühungen um eine weitergehende Standardisierung, um die Integration in Gruppenarbeitsplatzumgebungen und um weitergehende Fähigkeiten im Bereich multimedialer Kommunikation spiegeln den hohen Stellenwert wider, der diesen Systemen zugesprochen wird [Reinhardt 93].

Anforderungen an Mailsysteme lassen sich sowohl aus dem individuellen Gebrauch von elektronischer Post ableiten, als auch — und das ist wichtiger — aus der Funktion, die das Kommunikationsmedium für die Zusammenarbeit für mehrere Personen hat. [Mackay 88] analysiert das individuelle Verhalten von Mail-Nutzern und identifiziert Strategien, die Informationsflut zu handhaben. Hier sind Funktionen wie die Bewertung der Priorität von Briefen und ihre Archivierung von Bedeutung. Hinsichtlich der Kommunikationsfunktion von e-mail lassen sich zwei Aspekte unterscheiden:

- Die Distributionsfunktion betrifft einen einzelnen Kommunikationsschritt. Hier ist zu unterscheiden zwischen Briefen, die sich an einzelne Personen oder an Gruppen richten. Weiterhin können durch elektronische Post nicht nur Texte, sondern durch sogenannte 'Attachments' auch multimediale Daten übertragen werden (s. z.B. [Postel et al. 88]). Im Extremfall können via e-mail auch Programme übertragen und zur Ausführung gebracht werden [Borenstein 92]
- Die Konversationsfunktion bezieht sich auf die Intention der Kommunikation und auf den Zusammenhang zwischen mehreren Kommunikationsschritten. Das bedeutet im einfachsten Fall, daß ein Netzwerk der gegenseitigen Bezugnahme zwischen den archivierten Briefen aufgebaut wird und Funktionen zur Fortsetzung von Konversationen angeboten werden [Goldberg et al. 92]. Weitergehend kann aber auch die Art der Bezugnahme erfaßt und sogar auch kontrolliert werden. Als Kommunikationsmodell dient in solchen Systemen die Sprechakttheorie als Diskursmodell für die Konversation (s.a. Abschnitt 2.2 in Teil I). Durch spezielle Attribute wird einem elektronischen Brief eine konversationale Funktion zugesprochen, die sowohl den Autor als auch den Empfänger des Briefs auf bestimmte Handlungen festlegt. Ein Vertrag z.B. ist eine verbindliche Selbstverpflichtung des Autors, die vom Empfänger Einverständnis verlangt. Eine Konversation besteht dann aus dem wechselseitigen Austausch von Briefen, die jeweils bestimmte Handlungen verlangen. Manche dieser Systeme erzwingen die Spezifikation von Sprechakten für jeden Konversationsschritt und überprüfen die Korrektheit der Sprechaktypen (z.B. Coordinator [Flores et al. 88]), während andere eine derartige Strukturierung der Konversation optional anbieten, wie z.B. Strudel [Shepherd et al. 90].

3.2.2 E-mail im KHS

Die Integration von elektronischer Post als spezielle KHS-Applikation erfolgt, wie bei anderen Anwendungen auch, durch eigene Knoten- und Verknüpfungstypen, die hier eingeführt werden, bevor auf Aspekte der Interaktion mit Mail eingegangen wird.

3.2.2.1 Objekttypen

Elektronische Briefe werden durch semistrukturierte Knoten dargestellt, die neben einem freien Textteil formatierten Zugriff auf die standardisierten Attribute elektronischer Briefe, Adressaten (s. RFC 822) — direkter Adressat, Kopie (cc), verdeckte Kopie (bcc) —, Absender und Thema, erlauben. Zusätzlich sind Felder vorgesehen für einen Brieftyp, der Auskunft über den Zweck des Briefs gibt (z.B. Memo oder Termin Vereinbarung) und für die Aktion, die vom Empfänger des Briefs erwartet wird (z.B. 'baldige Antwort' oder 'eilige Bearbeitung'). Freie, vom Nutzer zugewiesene Attribute werden beim Versenden

elektronischer Post wie Standardattribute behandelt und erlauben so die Anpassung an externe Mailsysteme, die weitere Attribute erfassen können.

Die Archivierung von elektronischer Post erfolgt nach den in Abschnitt 3.1 beschriebenen Verfahren normalerweise in speziellen Strukturknoten, sogenannten 'Mailfoldern', allerdings kann Mail auch in andere Strukturknoten, z.B. typübergreifende Dossiers eingeordnet werden. Spezielle Mailfolder wiederum sind darauf eingerichtet, Listserver verwalten zu können. Ihnen kann eine Adreßliste zugeordnet werden, an die alle eintreffende Mail automatisch (unredigierter Server) oder nach Bestätigung durch einen Moderator (redigierter Server) weitergeleitet wird. Derartige Polder können fest vorgegebene Mailformate zur Verwaltung des Servers (subscribe, unsubscribe) automatisch verarbeiten.

Konversationen über mehrere Kommunikationsschritte hinweg werden durch zwei Formen der Bezugnahme unterstützt:

- Durch Bezug auf einen eindeutigen Identifikator (Message-Id) kann ein Brief als Antwort auf einen anderen definiert werden.
- Durch vollständiges oder partielles Einschließen des Texts eines anderen Briefes kann auf diesen Bezug genommen werden.

Derartige Beziehungen zwischen Briefen werden im KHS durch Verknüpfungen repräsentiert, deren Typ die Art der Bezugnahme wiedergibt ('reply-to', 'includes'). Mit Hilfe von e-mail können, wie im MIME-Standard festgelegt, auch Dateien mit nicht-textuellen Daten als sogenannte 'Attachments' übertragen werden. Die Beziehung zwischen transportierendem Mailobjekt und übertragener Information wird im KHS wiederum durch einen eigenen Verknüpfungstyp ('attached-to') repräsentiert.

3.2.2.2 Empfangen und Versenden von Mail

Die Entgegennahme von Mail wird normalerweise durch einen eigenständigen Agenten betrieben, der parallel zu der normalen Nutzerinteraktion mit dem Hypertext operiert. So ist eine zeitaktuelle Einordnung von Mail möglich. In regelmäßigen Abständen stellt der Agent eine Verbindung zum Mailsystem des Betriebssystems her und erfragt von ihm einzeln alle zuletzt eingetroffenen Briefe. Jeder Brief durchläuft dann die folgenden Bearbeitungsschritte:

- Die strukturierten Felder werden erkannt, deren Inhalte werden einem neu erzeugten Mailknoten als Attribute zugeordnet.
- Anhand der Message-Id wird überprüft, ob der Knoten schon im Hypertext enthalten ist. Wenn ja, wird die weitere Bearbeitung abgebrochen.
- Ist der Brief als Antwort auf einen anderen im Archiv vorhandenen ausgewiesen, so wird eine Verknüpfung erstellt.
- Der Text wird auf eingeschlossene Briefe und Attachments untersucht:
 - Eingeschlossene Briefe werden, wenn ihr Text einschließlich des formalen Teil vollständig ist, aus dem eigentlichen Text herausgelöst und als eigenständige Knoten repräsentiert, die zunächst den gleichen Analyseprozeß durchlaufen und dann an den umschließenden Brief durch Verknüpfungen angebunden werden.
 - Auch Attachments werden aus dem Text herausgelöst. Anhand der beigefügten Formatangaben wird erkannt, ob die Daten codiert und komprimiert sind und welches Kompressionsverfahren gewählt wurde. Gegebenenfalls werden sie decodiert und dekomprimiert. Es wird ein neuer Knoten erzeugt, dessen Typ

anhand des Typs des Attachments ausgewählt wird. Dieser Knoten wird in den Hypertext eingefügt (automatisch klassifiziert).

- Der neu erzeugte Mailknoten wird in den Hypertext eingefügt (automatisch klassifiziert).

Das Versenden von Mail verläuft analog. Durch ein spezielles Kommando erzeugt der Nutzer, evtl. als Antwort auf einen anderen Brief, einen neuen Mailknoten. In diesen trägt er die formalen Attribute und den eigentlichen Text ein. Durch Verknüpfungen wird spezifiziert, welche weiteren Briefe eingeschlossen und welche Knoten als Attachments angefügt werden sollen. Für die Attachments werden automatisch die adäquaten Codierungs- und Kompressionsverfahren gewählt.

3.2.2.3 Exkurs: Adressen

Eine für die tägliche Arbeit wichtige, wenn auch wegen ihrer geringen Komplexität nicht ausführlicher zu diskutierende Systemkomponente ist die Adreßverwaltung. Das KHS unterstützt sie durch einen Knotentyp, der Daten wie Name, Anschrift, Telefon etc. erfaßt. Diese Knoten werden dann zu alphabetischen Verzeichnissen aggregiert.

Im Zusammenhang der elektronischen Post wird die Adreßverwaltung als ein einfaches Beispiel für die integrierte Nutzung von Informationsressourcen relevant. Jede Adresse enthält neben der e-mail-Adresse auch ein individuelles Kürzel, das beim Versenden als Alias verwendet werden kann. Wird ein elektronischer Brief im Kontext eines Adreßknotens erzeugt, so wird die e-mail-Adresse automatisch übertragen.

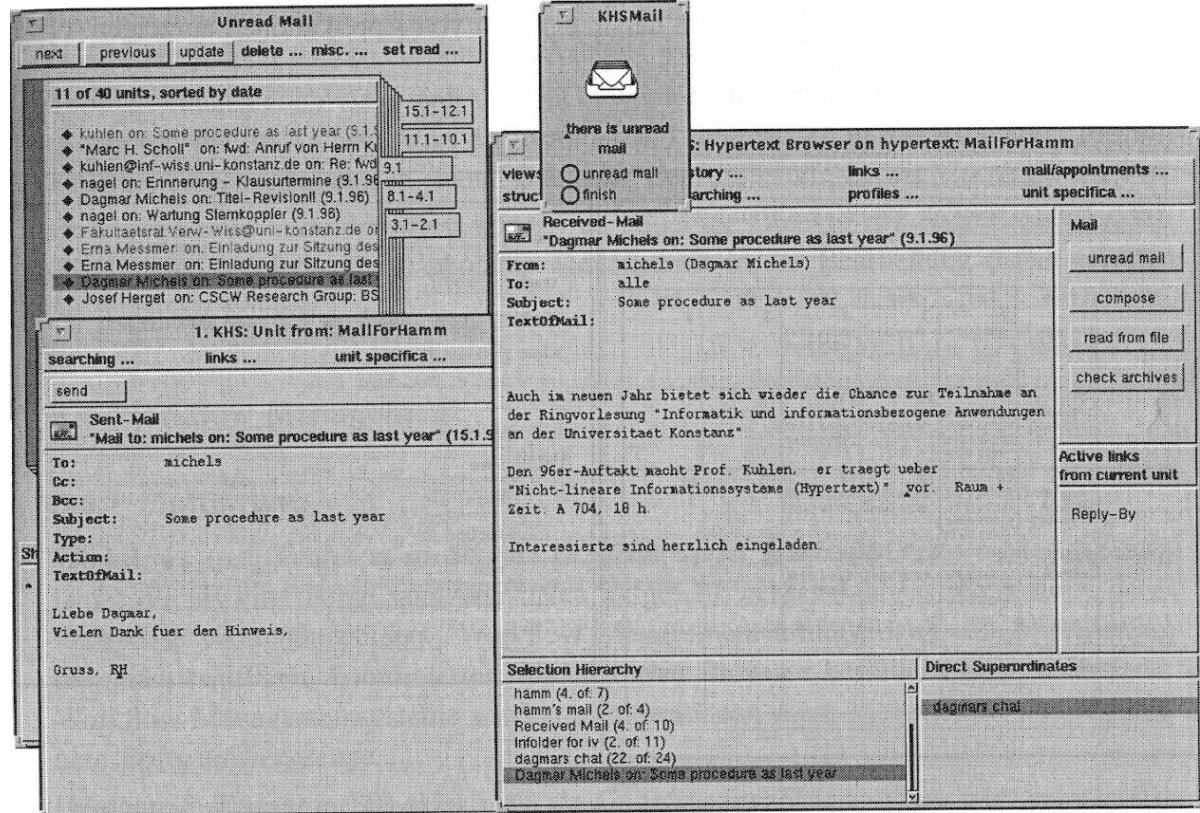


Abbildung 13 Typische Umgebung für die Arbeit mit e-mail — in einem gesonderten Werkzeug liegt eine Liste der bisher ungelesenen Briefe vor. Diese können nacheinander angewählt und im Hypertextbrowser gelesen werden. Soll eine Antwort verfaßt werden, so wird ein gesondertes Werkzeug zur Bearbeitung des neuen Briefs geöffnet. Der aktuelle Status des Mailsystems — ist neue oder ungelesene Mail vorhanden — wird in einem eigenen Fenster präsentiert, dessen Entwurf sich an der Ikonographie Mailsysteme orientiert.

Beim Versenden und Empfang von Briefen werden automatisch Verknüpfungen zwischen dem Absender und Empfänger eines Briefs und dem zugehörigen Adreßknoten erstellt, so daß über die Adressen leicht eine Korrespondenz mit einer Person aufgerollt werden kann.

3.2.2.4 Interaktion mit Mail

Die Interaktion mit dem Mailsystem erfordert keine eigenen Werkzeuge. Vielmehr müssen die im KHS generell vorhandenen Werkzeuge von den neu definierten Knotentypen nur adäquat parametriert werden:

- Editieren von Mail geschieht mit dem normalen Editor.
- Suchen von archivierter Mail erfolgt üblicherweise durch Formularsuche oder Navigieren in der Archivstruktur.
- Mailspezifische Funktionen, wie Suche nach ungelesener Mail, werden durch typspezifische Funktionen basierend auf Standardwerkzeugen unterstützt.

3.2.2.5 Zusammenfassung und Ausblick zum Thema e-mail

Die Einbindung von e-mail in Hypertext hat — hier erstmalig in einer wirklichen Anwendungssituation — den modularen Aufbau des Hypertextmodells des KHS bestätigt, indem wichtige Funktionen wie z.B. die Archivierung applikationsübergreifend spezifiziert werden konnten. Aus diesem hohen Grad an Integration ergibt sich auch der hervorstechendste Vorteil des Mailsystems, wie es im KHS realisiert ist, gegenüber Standardtools. Der Preis ist eine komplexere Systemumgebung mit einer noch nicht so bedarfsgerecht zugeschnittenen Funktionalität. Eine vorläufige Evaluation, welche diese Einschätzung stützt, gibt [Spribile 95].

In den weiteren Beispielen werden wir die Existenz einer Mailkomponente als gegeben annehmen und weitere Möglichkeiten der Integration von Applikationen aufzeigen.

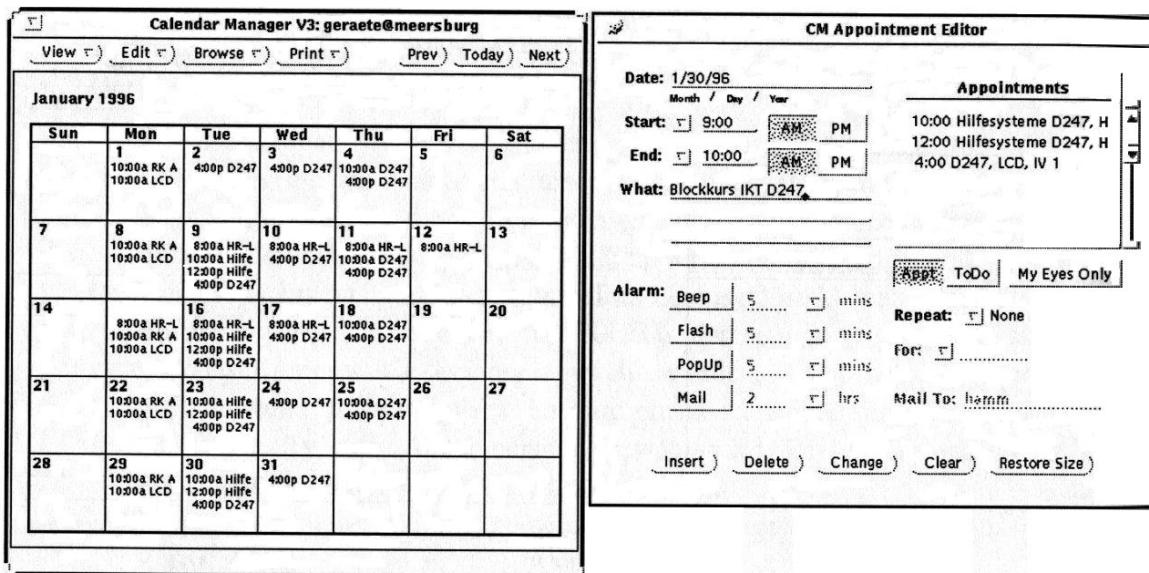


Abbildung 14 Schnittstelle des Sun-OS Calendar-Managers— neben der dargestellten Monatsübersicht können auch Übersichten über das Jahr, die Woche und den Tag präsentiert werden.

3.3 Kalender

Ein zentraler Gesichtspunkt der Zusammenarbeit innerhalb von Gruppen ist die gemeinsame Terminplanung und -Vereinbarung, da viele Formen der Zusammenarbeit durch vereinbarte Termine gesteuert werden, die Vereinbarungen kommen wiederum durch Kommunikation von Terminvorschlägen eventuell mit Unterstützung von Terminplanungssystemen zustande.

Die Terminverwaltung, wie sie zur Zeit in das KHS integriert ist, betont diesen integrativen Aspekt als methodisch reizvolle Fragestellung, da hier wiederum die Stärken des Hypertextansatzes betont werden können. Demgegenüber treten aus Zeitgründen andere Fragen in den Hintergrund. Im Gegensatz zu elektronischer Post z.B. würde die Terminplanung eine eigene, einer Kalendermetapher folgende Nutzerschnittstelle benötigen (vgl. Abb. 14), die für das KHS bisher aber noch nicht realisiert wurde. Andererseits erweisen sich die listenorientierte Übersichtsinstrumente des KHS auch für die Verwaltung von Kalendern als relativ geeignet.

Wie elektronische Briefe lassen sich auch Termine nach formalen Kriterien klassifizieren. Die Grundlage für die Strukturierung eines Informationsraums für die Terminverwaltung sind z.B. der Gegenstand des Termins, der Zeitraum, der zur Verfügung steht, oder die zu dem Termin durchzuführenden Handlungen. Dementsprechend ist im KHS der Kern der Terminverwaltung wiederum ein gemeinsames Archiv, das Bereiche für gemeinsame, gruppenspezifische und private Termine einräumt. Termine werden durch Formulknoten repräsentiert, die strukturierten Zugriff zu den oben genannten Attributen erlauben und darüber hinaus festlegen, ob und mit welcher Frequenz ein Termin wiederholt wird (täglich, wöchentlich, montags, monatlich, ...). Die Überprüfung der Termine und die Aktivierung von Vorwarnfunktionen wird von einem eigenständigen Agenten übernommen. Anstehende Termine werden z.B. durch Mail bekanntgegeben, es können jedoch andere Aktionen vereinbart werden.

Termine können auch via e-mail verschickt werden. Je nach Kompetenz des Versenders (Schreibrechte im Archiv), wird der Termin dann automatisch in den passenden Terminkalender eingetragen oder in einen Mailfolder, der für Terminabsprachen vorgesehen ist.

Durch spezielle Verknüpfungen können an beliebige Objekte des Hypertexts Termine angebunden werden, um so geplante Meilensteine in der Bearbeitung dieser Objekte festlegen zu können.

3.4 Workflow Management

3.4.1 Einführung

Workflows (s. [Teufel et al. 95, S. 182-208] oder [Jablonski 95]) sind Arbeitsabläufe, die als eine Folge von Aktivitäten aufgefaßt werden können, die jeweils durch Ereignisse veranlaßt werden. Workflow Management umfaßt dann alle Maßnahmen, die einen regulären, koordinierten Ablauf dieser Prozesse befördern oder garantieren. Workflow Managementsysteme wiederum unterstützen diese Maßnahmen durch geeignete Softwarewerkzeuge.

[Malone & Crowston 90] skizzieren als *Coordination Theory* einen interdisziplinären Rahmen für derartige Koordinationsprozesse. Danach kann Koordination sich an unterschiedlichen Komponenten manifestieren, die jeweils eigene Koordinationsvorgänge mit sich bringen (s. Tab. 15). Wir wollen im folgenden von einer wohlstrukturierten Problematik mit vorgegebenen Zielen und entsprechenden Aktivitäten ausgehen. Es ist also eine Zuordnung von Aktivitäten zu Personen unter Berücksichtigung von Abhängigkeiten erforderlich.

[Teufel et al. 95, S. 188] unterscheiden vier Modelle der Koordination:

- **Formularorientierte Modelle** erzielen Koordination durch Zirkulation eines Dokuments, das ist meist ein Formular, das von mehreren Stellen bearbeitet wird. Regeln für die meist sequentielle Bearbeitungsfolge und die auszuführenden Einzelschritte sind dem Dokument beigeordnet. Der Versand der Dokumente erfolgt per e-mail.

- **Vorgangsorientierte Modelle** beruhen auf einer logisch zentralen Modellierung von Vorgängen, z.B. durch Petri-Netze oder verwandte Formalismen.
- **Kommunikationsorientierte Modelle** basieren auf der Annahme, daß Koordination Kommunikation erfordert. Dementsprechend werden kommunikative Verbindungen durch Kommunikationsgraphen modelliert, die zur Steuerung des Transports von Dokumenten dienen.
- **Konversationsorientierten Modellen** liegt die Sprechakttheorie zugrunde [Wingrad & Flores 86]. Sie modellieren die Intentionen der Kommunikationspartner explizit durch Sprechakte und können somit nicht nur den Verlauf, sondern auch die Ziele der Kommunikation erfassen und steuern. Diese Kontrolle wird von den Nutzern jedoch häufig als hinderlich empfunden und umgangen [Bullen & Ben-nett 90].

Komponenten der Koordination	Assoziierte Kontrollprozesse
Ziele	Ziele benennen
Maßnahmen	Ziele auf Maßnahmen abbilden
Handelnde	Handelnde auswählen
Abhängigkeiten	Management der Abhängigkeiten

Abbildung 15 Komponenten der Koordination und ihre Koordinationsprozesse nach [Malone & Crowston 90]

3.4.2 Workflow Management mit Hilfe des KHS

Das Modell des Workflow Management innerhalb des KHS ist in erster Linie dokumenten- oder formularorientiert, eine Ausrichtung, die für ein Hypertextsystem naheliegt. Es beinhaltet allerdings auch gewisse kommunikationsorientierte Aspekte. Grundsätzlich erfolgt die Steuerung von Workflows anhand der Struktur des Informationsraums, indem Dokumente innerhalb dieser Struktur bewegt werden. Durch die Einordnung eines Dokuments erfolgt die Zuordnung von Akteuren und erforderlichen Aktivitäten.

Die eigentlich steuernden Einheiten sind spezielle Strukturknoten — im folgenden WFF (Workflowfolder) genannt, die jeweils als Durchlaufstation für Dokumente dienen, an denen dabei jeweils ein Arbeitsschritt vorgenommen wird. Dieser Arbeitsschritt kann im Extremfall automatisch erfolgen, wird in der Regel jedoch den Eingriff eines Nutzers erfordern. Der Ablauf eines Workflows innerhalb des KHS ist folgendermaßen zu beschreiben:

- Ein zu bearbeitendes Dokument wird via e-mail oder durch eine Nutzeraktion in das System eingebbracht.
- Das Dokument wird klassifiziert und in die Struktur eingeordnet. Dabei erfolgt eine Einordnung in das Archiv und weiterhin in einen WFF, einen aufgabenspezifischen Strukturknoten.
- Dem WFF ist über eine spezielle Verknüpfung ein Frame zugeordnet, der Angaben über den zuständigen Sachbearbeiter enthält. Sind mehrere Sachbearbeiter für gleichartige Vorgänge verantwortlich, so wird jedem ein eigener WFF zugeordnet. Die Auslastung wird dann über die Zuteilungsregeln gesteuert.
- Andere Verknüpfungen ordnen dem WFF Aufgabenbeschreibungen zu, die, falls sie regelhaft formuliert sind, auch automatisch ablaufen können. In diesem Fall haben die Bearbeiter nur Überwachungsfunktion. In der Regel wird die Aufgabe in ihrem inhaltlichen Teil nur verbal spezifiziert. Formale Angaben betreffen z.B. den Zeitrahmen, in dem die Aktivität abgeschlossen sein muß.

- Beim Einfügen des Dokuments in den WFF wird automatisch ein Knoten erzeugt, der eine Terminangabe repräsentiert. Dieser wird in das Terminverzeichnis des Sachbearbeiters eingetragen und durch eine Verknüpfung mit dem Dokument verbunden. Ein dem Termin zugeordneter Kommentar gibt eine Beschreibung der Aufgabe, benennt den Bearbeiter und den Bearbeitungszustand. Ist eine ganze Arbeitsgruppe vom Arbeitsfortschritt entscheidend betroffen, können die Termine auch zusätzlich in einen globalen Bereich eingetragen werden, so daß ständig ein Feedback über den Stand der Arbeit gegeben ist (s.a. [Dourish & Bellotti 92]).
- Der Abschluß eines Arbeitsschrittes wird durch den Nutzer explizit mittels eines Kommandos definiert. Als Folge wird zunächst die zugeordnete Terminangabe so modifiziert, daß als Endtermin anstelle des bisher eingetragenen Soll das Ist notiert wird, als Bearbeitungszustand wird 'beendet' eingetragen. Alternativ ist eine erfolglose Beendigung möglich, die als Abbruch notiert wird.
- Als Folge des erfolgreichen Abschlusses eines Arbeitsschritts muß das Dokument zur nächsten Bearbeitungsstation weitergeleitet werden. Dies kann auf zweierlei Art geschehen;
 - Durch die Bearbeitung wurde das Dokument verändert. Ein erneuter Klassifikationsschritt führt zu einer Einordnung in einen Strukturknoten, der für einen anderen Verarbeitungsschritt zuständig ist. Reihenfolgen in der Bearbeitung ergeben sich nur aus inhaltlichen Vorbedingungen, welche die Dokumente erfüllen müssen. Wird ein Dokument mehreren WFFs gleichzeitig zugeordnet, so ist eine Parallelverarbeitung möglich. Nachteil dieses Verfahrens ist die verteilte und deshalb nicht leicht zu wartende Regelstruktur. Bei weitgehend unabhängigen Regeln wirkt sich die Verteilung allerdings zum Vorteil aus, indem keine Abfolgen explizit modelliert werden müssen.
 - Alternativ können die Transportverbindungen zwischen Mitarbeitern auch explizit durch spezielle Verknüpfungen, sogenannte Workflowlinks (WFL) modelliert werden. Verbindet ein WFL zwei WFFs, so wird ein Dokument nach Beendigung eines Arbeitsschritts entlang dieser Verknüpfung von einem WFF zum anderen transportiert. Besitzt dieser untergeordnete Strukturknoten, so treten wiederum Klassifikationsregeln in Kraft (s.a. Abb. 16). Dieses Verfahren bietet zunächst den Vorteil einer expliziten Modellierung der Kommunikationswege. Darüber hinaus setzt es nicht zwingend voraus, daß alle Arbeitsschritte vom KHS kontrolliert ablaufen. Anstelle eines WFF kann nämlich auch ein Adreßknoten als Ziel eines WFL dienen. In diesem Fall erfolgt der Dokumenttransport via e-mail. Anhand der Mailadresse des Adreßknotens wird ein Mail-Knoten erzeugt, der als Text Aufgabenbeschreibung und Termine enthält. Diesem wird als Attachment das Dokument angefügt und zusammen mit ihm abgesandt. Workflow Management via Mail setzt allerdings im allgemeinen einen sequentiellen Ablauf voraus. Nur wenn sich Arbeitsschritte auf getrennte Teile eines wohlstrukturierten Dokuments beziehen, dessen neu entstehenden Versionen später automatisch identifiziert und zusammengeführt werden können, ist Parallelverarbeitung via Mail möglich.

3.4.3 Exkurs: Verwaltung einer elektronischen Zeitschrift mit dem KHS

Die dynamische Entwicklung der neuen, elektronischen Kommunikationsmedien hat ein neues Publikationsprodukt hervorgebracht, die elektronische Zeitschrift. [Kuhlen 95, S. 505-511] erklärt die rasche Zunahme wissenschaftlicher Zeitschriften, die in elektronischer Form publiziert werden, aus folgenden Motiven:

- Die zunehmende fachliche Spezialisierung bringt Fachorgane mit derartig geringer Auflage hervor, daß kein Verlag das Risiko übernehmen mag. Die Herausgabe elektronischer Zeitschriften in Eigenregie der Wissenschaftler ist dann als Selbsthilfe zu begreifen.
- Auch für die konventionellen Printmedien werden den Autoren häufig druckfähige Vorlagen abverlangt. Diese sind zumeist schon in einer Form, welche die Übertragung auf ein elektronisches Medium nahelegt.
- Vor allem niveauvolle gedruckte Zeitschriften publizieren Artikel erst nach einer langen, durch Begutachtungsvorgänge und langfristige Publikationspläne bedingten Vorlaufzeit, die im Zusammenhang mit dem ständig steigenden Bedarf an Aktualität nicht mehr zu vereinbaren ist. Von elektronischen Publikationen verspricht man sich eine Vereinfachung und Beschleunigung der Begutachtungsprozesse.
- Weitere erwartete Mehrwerte betreffen die weltweite Distribution, die verbesserten Suchmöglichkeiten und die einfache Integration in die mittlerweile übliche rechnergestützte Arbeitsumgebung des Wissenschaftlers.

Gerade der Erstellungsprozeß einer elektronischen Zeitschrift (in leicht idealisierter Form) scheint als angemessen, um die Möglichkeiten von Hypertext zur Unterstützung von Workflows zu illustrieren, da hier Planungs- und Publikationsinstrument zusammenfallen. Einen hohen Anteil am Verwaltungsaufwand einer Zeitschrift nimmt der Begutachtungsprozeß ein. Eine effiziente Unterstützung gerade dieses Vorgangs ist essentiell (vgl. [Diaper 95]), da hier ein erhebliches Potential zur Steigerung der Aktualität liegt. Da renommierte Autoren erst noch von den Standards des neuen Mediums überzeugt werden müssen, darf jedoch nicht der Eindruck entstehen, daß die Gründlichkeit der Begutachtung und damit das Niveau der Zeitschrift durch diese Effizienzsteigerung litt.

Im folgenden wird der Entstehungsprozeß einer elektronischen Zeitschrift skizziert und aufgezeigt, wie er durch Hypertext unterstützt werden kann:

- Der Vorgang beginnt mit dem Empfang eines via e-mail verschickten Artikels in der Zeitschriftenredaktion. Der Artikel, dem elektronischen Brief als Attachment beigefügt, wird automatisch in einen Eingangsordner gestellt, der Autor erhält eine Eingangsbestätigung.
- Das Dokument wird automatisch in ein hypertextkonformes Format konvertiert und in Hypertextknoten segmentiert und zusätzlich mit einer Formalbeschreibung versehen. Das Ergebnis wird intellektuell überprüft und ergänzt. Das Dokument wird in einen eigenen Polder eingefügt, der weiterhin mit dem Dokument alle weiteren Bearbeitungsschritte durchläuft. Hier werden alle Versionen des Dokuments, alle Kommentare und Annotationen gesammelt. Im Dokument enthaltene Literaturangaben werden mit der dem Journal zugeordneten Zentralbibliographie (s. Abschnitt 3.5) abgeglichen, Literaturzitate durch Verknüpfungen repräsentiert.
- Zu Beginn der Begutachtungsphase wird das Dokument in einen Hypertextbereich eingestellt, der von allen Herausgebern und Referees eingesehen werden kann. Daß jeder aus diesen Gremien Kommentare zu allen eingereichten Artikeln abgeben kann, trägt zur Objektivierung der Begutachtung bei.
- Zusätzlich wird das Dokument anhand der thematischen Auszeichnung und evtl. weiterer Kriterien einer Anzahl von Referees zugeteilt. Jeder der Referees, die kontinuierlich mit der Zeitschrift kooperieren, verfügt ständig über seinen eigenen Arbeitsbereich im Hypertext, aber auch externen Referees wird temporär ein Ar-

beitsbereich zugeteilt. In diesen wird der Artikel klassifiziert, eine Terminangabe wird dem Dokument zugeordnet.

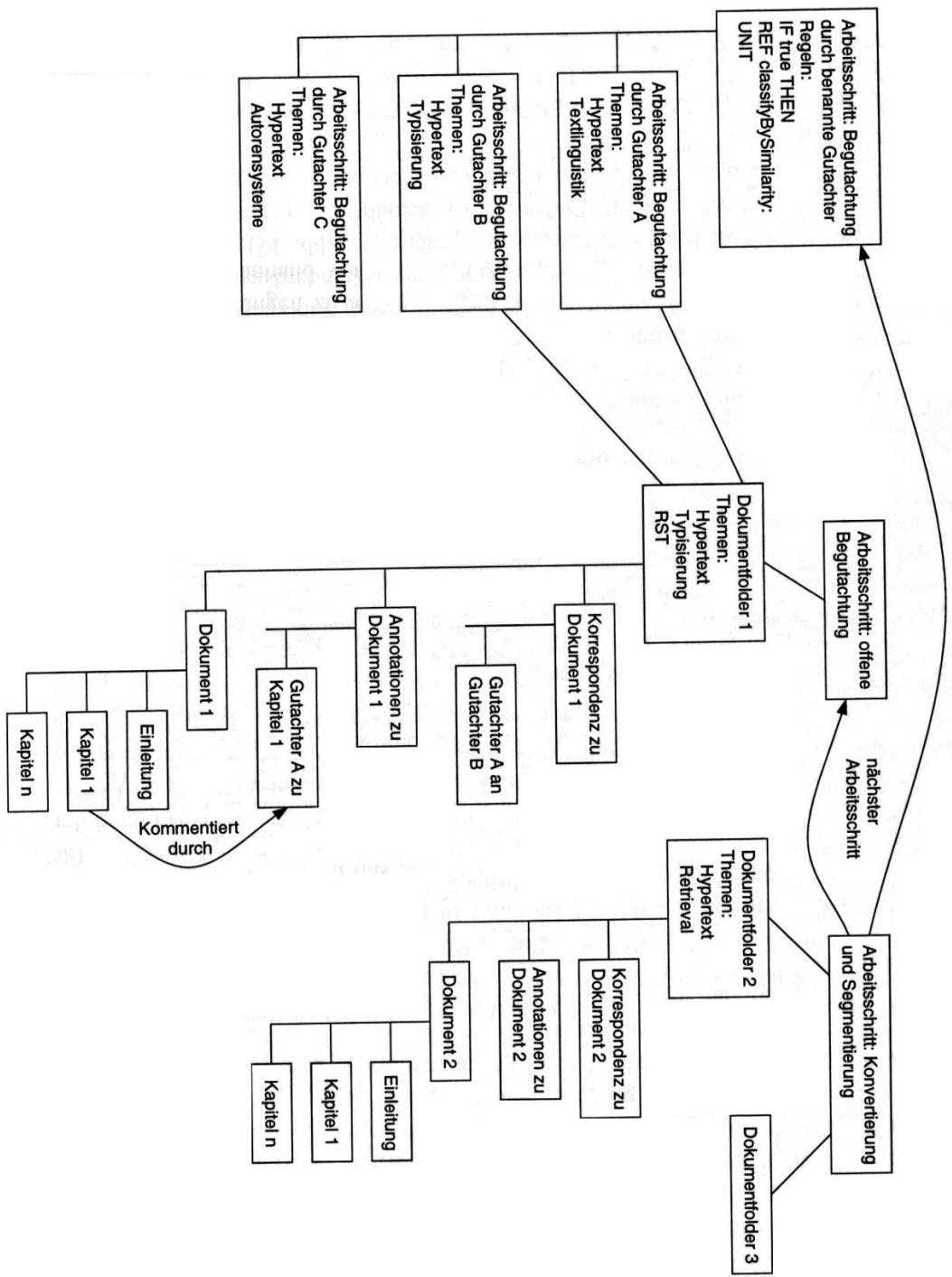


Abbildung 16 Arbeitsschritt aus dem Refereeprozeß für elektronische Zeitschriften

- Die Gutachter können das Dokument mit KHS-Werkzeugen oder auch mit einem WWW-Browser (z.B. Netscape) anschauen. Ihre Kommentare fügen sie dem Dokument als Annotationen bei. Diese Annotationen sind auch den anderen Gutachtern zugänglich und wiederum annotierbar. Eine fachliche Debatte kann auch via e-mail geführt werden. Die entsprechenden Briefe werden automatisch dem zu begutachtenden Dokument zugeordnet.
- Nach Abschluß der Begutachtung wird das Dokument wieder in einen Arbeitsbereich überführt, für den ein Redakteur zuständig ist. Dieser überprüft kurz, welche

Kommentare dem Autor zugänglich gemacht werden sollen und setzt die Zugriffsrechte entsprechend, so daß der Autor Dokument und Kommentare einsehen kann.

- Wurde der Artikel endgültig abgelehnt, so wird das Dokument in ein internes Archiv eingefügt, ansonsten hat der Autor jetzt Gelegenheit, das Dokument zu bearbeiten, eine verbesserte Fassung einzureichen. Diese durchläuft evtl. die oben beschriebenen Schritte nochmals.
- Wurde der Artikel zur Publikation angenommen, erfolgt die Endbearbeitung:
 - Die textuelle Aufbereitung wird endgültig dem Erscheinungsbild des Journals angepaßt.
 - Verknüpfungen zu anderen Artikeln des Journals werden auf statistischer Basis oder anhand von formalen Ähnlichkeiten oder Koreferenzen automatisch erstellt.
- Zur Veröffentlichung wird das Dokument in einen Bereich des Hypertexts gebracht, welcher der Öffentlichkeit bzw. den Abonnenten zugänglich ist.

Mit der Veröffentlichung im WWW stehen dem Leser dann alle Navigations- und Suchoptionen des KHS zur Verfügung, Index- und Schlagwortsuche, Navigation in Dokumenten, Browsing anhand von Dokumentähnlichkeiten oder über Referenzen und Koreferenzen usw. Vor allem sind elektronische Zeitschriften im Gegensatz zu ihren gedruckten Vorgängern kein Medium, das seine Leser zu einer rein passiven Haltung verurteilt. Es ist vielmehr davon auszugehen, daß auch Leser die erschienenen Artikel kommentieren und annotieren können. Derartige Kommentare können dabei durchaus den Umfang von Koreferaten annehmen. Um einen Wildwuchs zu vermeiden, ist es vermutlich sinnvoll, die von Lesern eingehenden Beiträge zu ordnen. Unangemessene Beiträge müssen auch ausgeschlossen werden können. Diese Kontrolle sollte allerdings sehr großzügig ausgelegt werden. Um eine Diskussion anzuregen, kann es in manchen Fällen auch sinnvoll sein, die Annotationen der Gutachter z.T. als Kristallisierungspunkt für weitere Diskussion auch der Öffentlichkeit zugänglich zu machen. Eine elektronische Zeitschrift entwickelt sich so zu einem wissenschaftlichen Diskussionsforum, für dessen Verwaltung weitere Workflows vorzusehen sind.

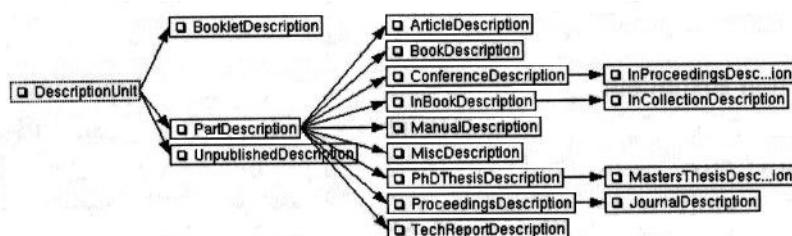


Abbildung 17 Typhierarchie für bibliographische Formularknoten

3.5 Literaturverwaltung

Arbeit mit Literatur und Bibliographien nimmt einen erheblichen Raum in der Arbeitszeit eines Wissenschaftlers ein. Wissenschaftliche Publikationen werden auch nach der Vollständigkeit und Korrektheit der beigefügten Bibliographie beurteilt. Erstrebenswert ist auch ein konsistenter Umgang mit Literatur in wissenschaftlichen Arbeitsgruppen vor allem natürlich dann, wenn es um kollektive Publikationen geht. Folgerichtig weisen [Musliner et al. 92] auf die Bedeutung bibliographischer Daten als Grundlage für kooperatives Arbeiten in wissenschaftlichen Gruppen hin. Allein um den Aufwand der Mehrfacherfassung zu vermeiden, lohnt sich eine gemeinsame Verwaltung von Literaturangaben. Des weiteren ist

die Möglichkeit aufgabenbezogener Ordnungsstrukturen für Bibliographien zu schaffen (Teilbibliographie für eine Publikation oder für einen Kurs).

Die Abbildung einer Bibliographie auf die Objektstruktur des KHS verläuft zunächst völlig analog zu den schon beschriebenen Anwendungen. Literaturreferenzen werden durch Formulknoten erfaßt und in ein Archiv von Strukturknoten eingeordnet. Hinsichtlich der zu erfassenden Attribute und der Definition von Subtypen zur Beschreibung spezieller Dokumenttypen wurde Bezug auf einen quasi-Standard genommen, nämlich auf die durch BIBTEX definierte Beschreibungssprache (s. Abb. 17). Die Nutzung von Bibliographien zur Zitierung innerhalb des KHS geschieht durch spezielle Verknüpfungen zwischen Text und Referenz, die anhand vorgegebener Zitiernormen automatisch erzeugt werden.

Es ist jedoch nicht davon auszugehen, daß gemeinsam verwaltete Bibliographien auch in einer einheitlichen Umgebung zur Texterstellung genutzt werden können oder sollen. Deshalb bietet das KHS die Möglichkeit, Bibliographien als BIBTEX-Dateien auszulagern und so externen Textverarbeitungsprogrammen in einem normierten Format zur Verfügung zu stellen (s. Abb. 18).

Ebenso müssen extern erstellte und recherchierte Bibliographien in den gemeinsamen Pool eingebracht werden können. Dementsprechend ermöglicht das KHS, aus BIBTEX-Dateien Bibliographien aufzubauen und so verteilte lokale Bestände zusammenzuführen. Häufig entstehen Bibliographien als Resultat von Online-Recherchen. Hier zahlt sich die Netzwerkoffenheit des KHS (s. Kapitel 7.1 in Teil II) insofern aus, als bibliographische Angaben aus Literaturdatenbanken automatisch in das KHS integriert werden können. Wegen der nicht immer konsistenten Erfassung von Texten in Datenbanken und der unterschiedlichen Erschließungstiefe müssen die entstehenden Hypertextknoten z.T. mit allerdings geringem Aufwand nachbearbeitet werden.

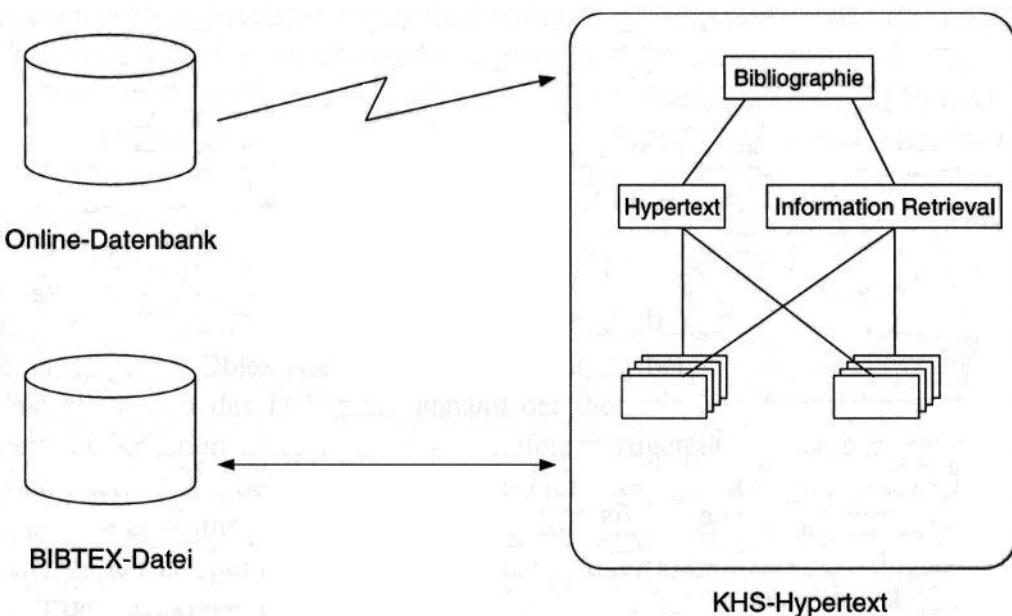


Abbildung 18 Austausch von bibliographischen Angaben zwischen Online-Datenbanken, dem KHS und BibTEX-Dateien

3.6 Zusammenfassung zum Thema computerunterstützte Gruppenarbeit

Die hier aufgeführten Beispiele konnten das umfangreiche Gebiet der computergestützten Gruppenarbeit selbstverständlich nur tangieren. Dennoch dürfte deutlich geworden sein, welche integrative Kraft in der gemeinsamen Nutzung strukturierter Arbeitsbereiche in Form

von Hypertexten liegt, insbesondere, wenn durch Typisierung von Objekten eine enge Verzahnung von Struktur und Funktion erreicht werden kann.

4. Unterstützung der Software-Entwicklung durch Hypertext

4.1 Einführung

Der Entwurf von Software ist im Prinzip auch als eine spezielle Form der computerunterstützten Gruppenarbeit aufzufassen, deren Gegenstand die Erstellung eines besonderen Dokumenttyps — Programme — ist (s.a. [Boldyreff 94]). Die bisher eingeführten Verfahren zur Unterstützung der Gruppenarbeit, insbesondere die Kommunikation der Gruppenmitglieder und die Planung von Arbeitsschritten, sind für Softwareentwurf und -Wartung dementsprechend durchaus anwendbar. Vor allem die Bedeutung eines gemeinsam zugänglichen und verwalteten Informationsraums — in der Terminologie des Software-Engineering ist dies ein Repository —, der alle im Verlauf des Entwicklungsprozesses anfallenden Dokumente zueinander in Beziehung setzt und allen Projektmitgliedern zugänglich macht, wird in der Literatur hervorgehoben (z.B. [Schönthaler & Nemeth 90]).

In den letzten Jahren sind zahlreiche Methodiken des Software-Engineering entwickelt worden. Gemeinsam ist ihnen, daß Erstellung und Nutzung von Software als ein mehrphasiger Vorgang begriffen wird, in dessen Verlauf Dokumente unterschiedlicher Strukturierungstiefe und -art entstehen, die durch ein komplexes Netzwerk von Interdependenzen verbunden sind. Diese strukturelle Verwandtschaft zu Hypertextsystemen hat dazu geführt, daß Anwendungen aus dem Bereich des Software-Engineering schon früh in die Hypertext-Literatur eingegangen sind. Ein frühes Beispiel für die Unterstützung des Software-Engineering durch Hypertext ist DynamicDesign [Bigelow & Riley 87], das bereits als spezielle Anwendung der HAM erwähnt wurde. Alle Dokumente eines Softwareprojekts, von den Spezifikationen bis zu den Dokumentationen und zum Quellcode, werden in einem Hypertext verwaltet und zugänglich gemacht. Der Ansatz von DynamicDesign sieht Hypertext als das integrierende Dokumentenverwaltungssystem zwischen anderen Werkzeugen, z.B. Editoren, Compiler, Mailsystem [Bigelow 88]. Das von [Mössenböck 90] vorgestellte System She ist demgegenüber auf die Verwaltung des Quellcodes prozeduraler Programmiersprachen spezialisiert, stellt dafür allerdings auch die erforderlichen Editier- und Übersetzungsfunktionen zur Verfügung. Daß auch die Verwaltung der Dokumente durch ein umfassenderes Hypertextmodell besser unterstützt werden kann, demonstriert das System ELEN [Jarwa & Bruandet 90], das durch Einführung von Dokumenttypen eine weitgehend normierte Strukturierung des Informationsraums erreicht. Als weitere zentrale Eigenschaft von ELEN wird die Möglichkeit zur Versionierung angesehen, die in ihrer Ausdruckskraft den hier bereits eingeführten Standards entspricht (s. Abschnitt 3.2.3 in Teil II). Die Konsistenz zwischen verschiedenen Dokumenten, die denselben Gegenstand betreffen (z.B. Quell- und Objektcode derselben Funktion), kann durch Definition von Konsistenzregeln in einer eigenen Skriptsprache erreicht werden [Jarwa & Bruandet 92]. Während die meisten der hier vorgestellten Systeme ihre Stärken besonders in den späten Phasen des Software-Entwicklungsprozesses ausspielen, ist RETH [Kaindl 93] auf das Requirements-Engineering spezialisiert. Das System kombiniert wissensbasierte und hypertextorientierte Techniken zu einer semiformalen Methode.

Einen besonderen Aspekt des Software-Erstellung, nämlich die Wiederverwendung von Spezifikationen und Lösungen, beschreibt [Sedes 92]. Hier werden die besonderen Indexierungsmöglichkeiten eines Hypertextsystems (SYRIUS) genutzt, um den inhaltsorientierten Zugriff auf Software-Komponenten zu erleichtern. Der Zusammenhang zwischen Wiederverwendbarkeit und Software-Qualität, wie ihn [Karlsson & Morel 95] darlegen, läßt diesen Aspekt der Verwendung von Hypermediatechnik als sehr relevant erscheinen.

Die bisher angesprochenen hypertextorientierten Ansätze der Unterstützung des Software-Engineering sind weitgehend unabhängig von einer konkreten Modellierungsmethodik formuliert. Dies ist insofern unproblematisch, als Hypertexte als Netzwerke beliebig strukturierter Dokumente angesehen werden können, unproblematisch auch, als die frühen Phasen des Software-Entwurfs noch nicht auf eine konkrete Entwurfsmethodik zugeschnitten werden müssen. Soll der Entwurf allerdings auch innerhalb einer bestimmten Methodik und auf eine Zielsprache hin vom Hypertextsystem unterstützt werden, so muß dessen Modell entsprechend getypte Konstrukte für die Ausdrucksmittel der Methodik bzw. der Zielsprache bereitstellen und sich nicht, wie [Neubert & Oberweis 92] richtig konstatieren, in der bloßen Bereitstellung der Dokumente, so sinnvoll dies auch für Archivierungs- und Wartungszwecke sein mag, erschöpfen. Ein Beispiel für ein derartig hochstrukturiertes Hypertextsystem, das differenziert verschiedene Phasen des Software-Entwurfs und innerhalb einzelner Phasen z.T. noch mehrere Formalismen unterstützt, ist DIF [Garg & Scacchi 87], das Spezifikation durch Formulare vorsieht, die wiederum zu größeren Templates aggregiert werden können. Eine Erweiterung von DIF erlaubt die explizite Modellierung von Vorgehensweisen während des Software-Entwurfs.

Der Stellenwert, der dem Software-Engineering als einem unter mehreren Anwendungsbeispielen im Rahmen dieser Arbeit zugemessen wird, erlaubt nicht, dieses komplexe Fachgebiet in seinen Implikationen zu erschöpfen. Vielmehr wird im folgenden die Leistungsfähigkeit des Hypertextmodells punktuell an einer Phase des Software-Entwurfs erprobt. Ausgewählt wurde hier wie bei den meisten anderen Systemen auch die Implementation, also die Erstellung und Verwaltung des Quellcode mit der zugehörigen Dokumentation. Für diesen Bereich wurde ein operationaler Prototyp entwickelt. Als Zielsprache wurde hier Smalltalk und zwar der Dialekt Smalltalk-80 in Zusammenhang mit der Entwicklungsumgebung VisualWorks gewählt, da Smalltalk als Implementationssprache des KHS am leichtesten einzubinden war.

Daß auch die vorangehenden Phasen des Software-Entwurfs durch das KHS adäquat zu unterstützen sind, wurde im Rahmen seiner Diplomarbeit von Thomas Brodt [Brodt 94] belegt. Es wurden die Konstrukte (Templates) der Methode *Object Oriented Analysis and Design* (OOAD, [Booch 91]) auf Typen des KHS-Hypertextmodells abgebildet. Brodt hebt hervor, daß insbesondere die Flexibilität von Hypertext und die gute Unterstützung kooperativen Arbeitens das KHS als sehr geeignet für die Anwendung 'Software-Engineering' erscheinen lassen, daß andererseits die bislang fehlende Unterstützung der methodenspezifischen Diagrammtechniken den Nutzwert doch deutlich einschränkt.

Eine etwas modifizierte Perspektive ergibt sich, wenn man die Auffassung des Software-Entwurfs — die gilt vor allem für die frühen Phasen — von Ortner [Ortner 93] und Schienmann [Schienmann 96] einnimmt. Als zentraler Teil dieses sprachkritischen Ansatzes des Software-Engineering wird die Rekonstruktion einer unternehmensweiten Fachsprache aufgefaßt. Dazu gehört die Erhebung des Sprachgebrauchs in einer umfassenden Aussagensammlung, anhand derer dann eine kontrollierte Terminologie erstellt werden kann. Eine syntaktische Festlegung der Fachsprache erfolgt anhand relativ einfacher Satzbaupläne. Sind die Aussagen in die Fachsprache übertragen, können diese Ausdrücke, die ein geeignetes Kommunikationsmedium für den Anwender sind, automatisch in Spezifikationsstrukturen umgesetzt werden, die, basierend auf Diagrammtechniken, dem Entwickler vertraut sind. Die differenzierten Ausdrucksmittel eines offenen Hypertextsystems wie des KHS scheinen nun sehr geeignet zu sein, eine derartige Vorgehensweise zu unterstützen:

- Die Erhebungsformulare oder Interviews, die zunächst zur Erfassung des Sprachgebrauchs dienen, können als Formulare oder Medienobjekte in den Hypertext eingebbracht werden.

- Diese schwach strukturierten Hypertextknoten werden dann systematisiert und in normsprachliche Ausdrücke verwandelt. Verknüpfungen zeigen die Ableitungen an.
- Aus den normsprachlichen Ausdrücken wird ein Wörterbuch erzeugt. Diese Terminologie kann gut durch Wissensrepräsentationskonstrukte — im KHS durch Frames — dargestellt werden. Auch hier können durch Verknüpfungen die Definitionegrundlagen der einzelnen Begriffe nachvollziehbar gemacht werden.
- Aus formalen Repräsentationen der Terminologie können automatisch Diagramme für objektorientierte Spezifikationsmethodiken erzeugt werden.

Das vorangehend skizzierte Szenarium ist gewißlich stark vereinfacht, gibt aber dennoch das Potential des Einsatzes von Hypertexttechniken im Software-Entwurf korrekt wieder.

4.2 Hypertextbasierte Verwaltung von Smalltalk-Code durch das KHS

Wegen ihres hohen Strukturierungsgrades sind objektorientierte Programmiersprachen besonders für eine Unterstützung durch Hypertext geeignet, wie auch schon die Arbeiten von [Sametinger & Stritzinger 91] für C++ und [Ansellem 95] für Smalltalk-80 zeigen. [Osterbye 93] stellt eine interessante Beziehung zwischen hypertextunterstützter Entwicklung von Smalltalk-Programmen und dem von Knuth vorgeschlagenen *Literate-programming* her. *Literate-programming* [Knuth 84] ist ein Stil der Programmierung, der gut lesbare Programme hervorbringen soll. Das wird durch eine weitgehende textuelle Kommentierung — der Textteil überwiegt den Programmteil bei weitem — und eine aufwendige textuelle Aufbereitung der Programme erreicht. Das von Knuth vorgeschlagene Web-System ermöglicht dies durch die Kombination einer prozeduralen Programmiersprache (Pascal) mit T_{EX} als Textauszeichnungs- und Formatierungssystem. Außer durch die gute Lesbarkeit der Programme zeichnet sich Web durch die automatische Erstellung von Indexen aus, die das Auffinden relevanter Text-/Programmsegmente erheblich erleichtern. Laut Osterbye ist hypertextorientiertes Programmieren in der Tradition des *Literate-programming* zu verstehen, erlaubt aber eine verbesserte Strukturierung der Dokumente. Demgegenüber produziert das Web gegenüber den verfügbaren Hypertextsystemen immer noch die besser "handhabbaren" Papierausdrucke.

4.2.1 Smalltalk

Smalltalk [Goldberg 84, Lalonde & Pugh 90] ist eine objektorientierte Programmiersprache, deren Klassen in einer Monohierarchie angeordnet sind. Auch Klassen sind als Objekte zu verstehen, die Instanzen von Metaklassen sind. Dieser Umstand wird allerdings vor dem Programmierer verborgen, da Definitionen und Methoden von Klassen und Metaklassen in der Entwicklungsumgebung im Zusammenhang präsentiert werden. Jede Klasse wird durch eine Klassendefinition spezifiziert und kann durch einen Kommentar dokumentiert werden. Die Methoden einer Klasse können nach vom Entwickler zu bestimmenden Kriterien zu Kategorien zusammengestellt werden, ebenso können Klassen zu Klassenkategorien aggregiert werden. Damit wird ein Ordnungsschema eingeführt, das orthogonal zur Vererbungshierarchie ist.

Sowohl der Quell- als auch der semicompliierte Objektcode sind jeweils in einer großen zusammenhängenden Datei — das *Image* und die *Changes* — abgelegt. Mechanismen für die kooperative Erstellung von Code gibt es, sieht man für elementare Instrumente zum Austausch von Software zwischen 'Images' ab, nicht.

Die Entwicklungsumgebung umfaßt eine große Anzahl vordefinierter Klassen und Methoden, die durch eine graphische Nutzerschnittstelle zugänglich gemacht werden. Der Umfang dieser Bibliothek ermöglicht einerseits eine ungemein effektive Arbeit, erschwert

aber das Erlernen des Systems erheblich, da effiziente Such- und Orientierungsmöglichkeiten nicht gegeben sind.

In den folgenden Abschnitten wird erläutert werden, wie die Konstrukte von Smalltalk auf die Objekttypen von KHS abgebildet werden und welche Vorteile daraus entstehen.

4.2.2 Klassen

Struktur: Eine Klasse wird durch ein komplexes Template (s.a. Abb. 19) repräsentiert. Ein Strukturknoten enthält einen speziellen Textknoten mit der Strukturspezifikation von Klasse und zugehöriger Metaklasse sowie zwei Strukturknoten, die entsprechend den Methodenkategorien die Methoden der Klasse und der Metaklasse enthalten. Dabei können Methodenkategorien hierarchisch gestaffelt werden, so daß eine differenziertere Strukturierung der Methodenmenge möglich wird. Dies ist besonders bei generischen Klassen mit einer großen Anzahl an Methoden sinnvoll.

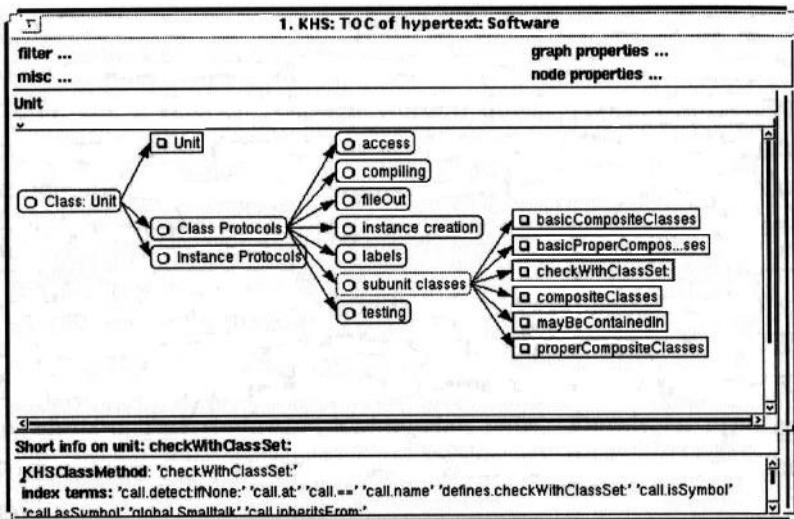


Abbildung 19 Hypertextorientierte Repräsentation einer Smalltalk-Klasse

Klassendefinition: Smalltalk unterscheidet zwischen einer Klassendefinition, die durch ein strikt formatiertes Template angegeben wird, und einem Klassenkommentar, der — weitgehend aus Freitext bestehend — keine zwingende Beziehung zur faktischen Systemstruktur hat. Dennoch wird ein bestimmtes Format für den Kommentar empfohlen, das Angaben über Zweck und Typ der Instanz- und Klassenvariablen vorsieht. Darauf aufbauend kann mit einem speziellen Werkzeug eine Konsistenzüberprüfung der Software vorgenommen werden. Im KHS wird der Unterschied zwischen Klassenspezifikation und -kommentar aufgegeben (s.a. Abb. 20). Vielmehr wird der formatierte Kommentar auch als Spezifikation aufgefaßt. So wird der Entwickler zur Angabe eines Minimalkommentars genötigt, eine Inkonsistenz zwischen Kommentar und Spezifikation kann nicht entstehen.

Gemeinsame Methoden: Smalltalk-80 unterstützt, zumindest in seiner Standardversion (siehe jedoch [Lalonde & Pugh 90, Bd. I, S. 237-239]), keine Mehrfachvererbung. Daraus resultiert, daß Klassenhierarchien nur nach jeweils einem Merkmal differenziert werden können, andere Eigenschaften müssen bei den Unterklassen häufig redundant spezifiziert werden. Das KHS bietet hier die Möglichkeit an, daß Methodenspezifikationen oder ganze Methodenkategorien von mehreren Klassen gemeinsam genutzt werden. Dies ist nur eine syntaktische Konvention, die besagt, daß der einem Methodenselektor zugeordnete Code in zwei oder mehreren Klassen gleich sein soll, entspricht aber nicht einer Mehrfachvererbung, so daß deren Konflikte und Idiosynkrasien vermieden werden (s.a. Abb. 21).

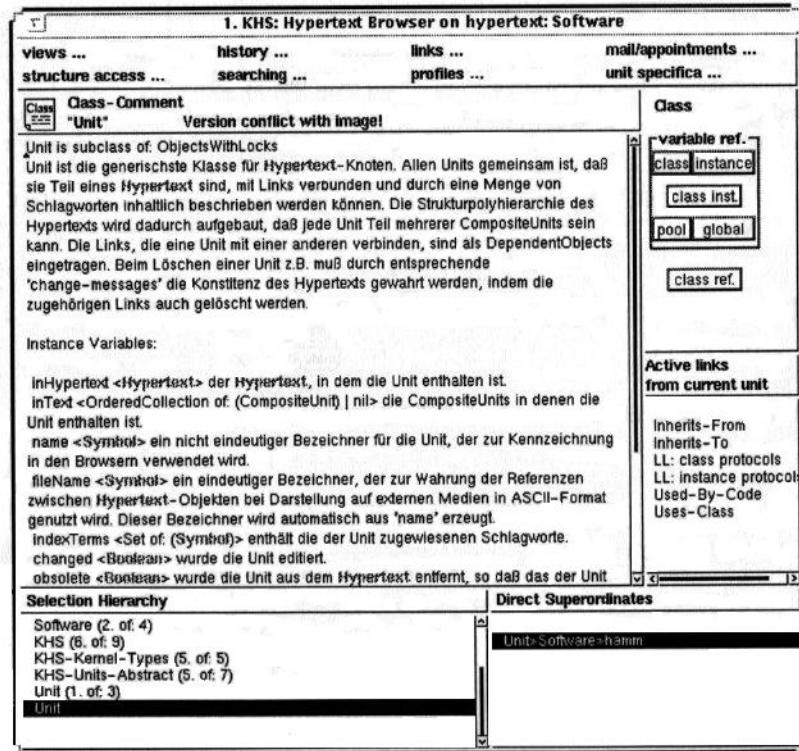


Abbildung 20 Klassendefinition— neben einem als Freitext angegebenen Kommentar ist eine formatierte Definition der Objektstruktur (Instanzvariable, Klassenvariable) gegeben. Eine Warnung gibt an, daß diese im Hypertext angegebene Spezifikation im Konflikt steht zu derjenigen, die compiliert im Image des Entwicklers vorliegt.

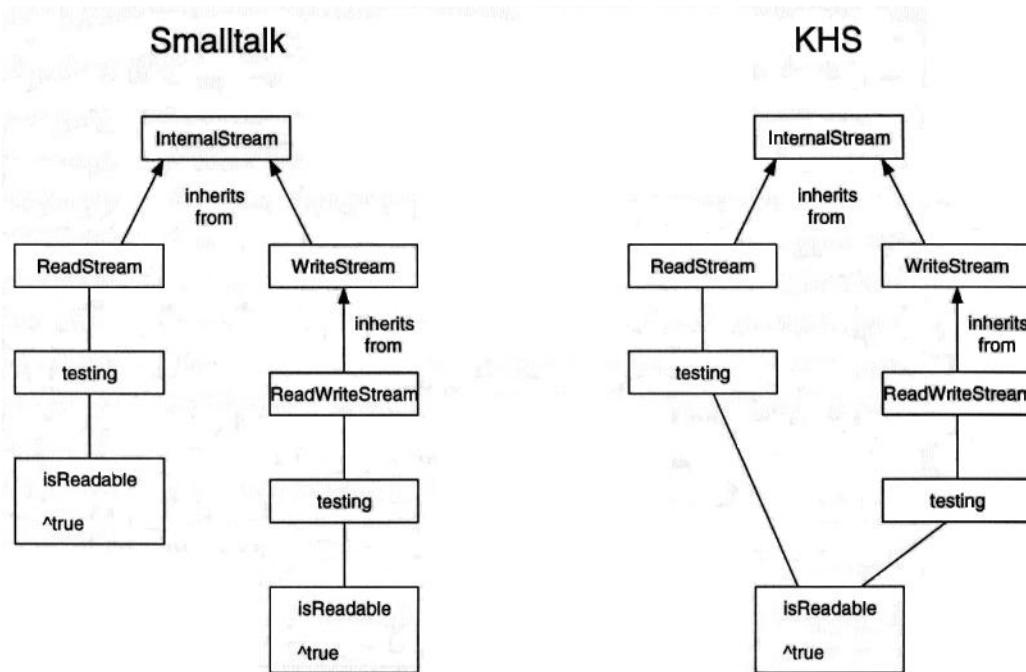


Abbildung 21 Gemeinsamer Code für Klassen— Smalltalk sieht interne Datenströme vor, die lesenden und/oder schreibenden Zugriff erlauben. Zunächst erfolgt die Aufteilung in Datenströme mit lesendem und schreibendem Zugriff. Datenströme, die beide Zugriffsarten erlauben, sind als Unterklassen zu Datenströmen mit schreibendem Zugriff definiert. Alle Methoden, die den lesenden Zugriff implementieren, müssen hier nochmals definiert werden, können im KHS aber gemeinsam verwaltet werden. Als Beispiel ist dafür die Methode 'isReadable' aus der Methodenkategorie 'testing' angegeben.

4.2.3 Ordnungsstrukturen

Das KHS stellt verschiedene Ordnungsstrukturen bereit, die bei der Orientierung im Software-Hypertext helfen sollen.

Kommentare und Annotationen Natürlich ist Smalltalk-Code annotierbar und damit kommentierbar wie jeder andere Hypertextknoten. Dabei werden Kommentare zu ganzen Klassen oder Methoden am günstigsten durch navigierbare Verknüpfungen angebunden, während Kommentare zu Implementationsdetails in Methoden am besten durch Stretchtext dargestellt werden. Indem so die Kommentare aus dem eigentlichen Code ausgeblendet werden können, vermindern auch lange Kommentare die Lesbarkeit nicht.

Als Gegenstand von Volltext-Suche sind Kommentare auch ein wichtiges Orientierungsmittel für die Software.

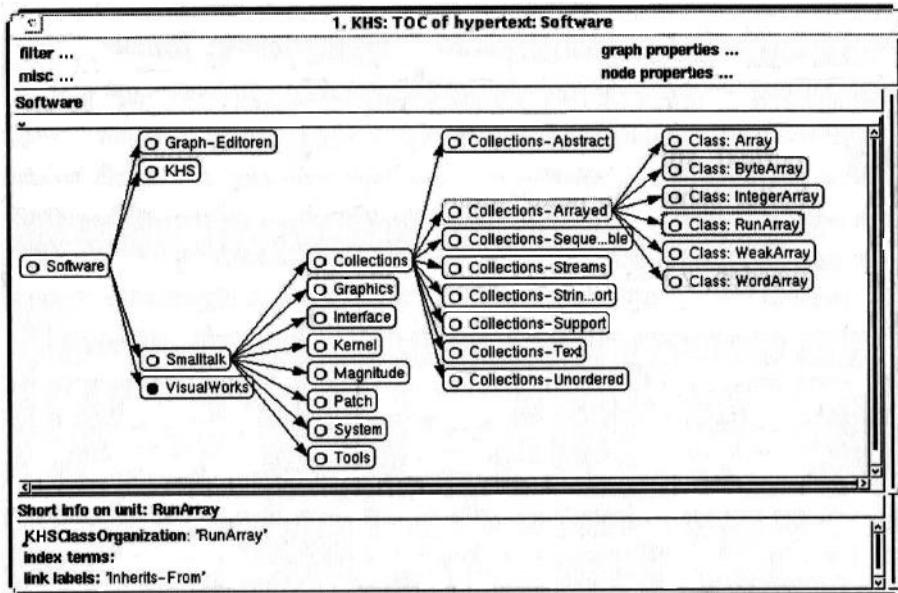


Abbildung 22 Ausschnitt aus der Kategorienstruktur eines KHS-Softwarehypertexts

Indexe: Wie jeder andere Hypertextknoten auch können Smalltalk-Klassen und -Methoden natürlich auch mit Schlagworten beschrieben werden. Hier wird allerdings ein Teil der Deskriptoren, die Eigenschaften des Codes wie Nutzung von Klassen oder Aufruf von Methoden erfassen, bereits automatisch vergeben.

Verknüpfungen: Vererbungsbeziehungen werden durch Verknüpfungen zwischen den Klassendefinitionen und zwischen den zugehörigen Strukturknoten repräsentiert. Auch Nutzungsbeziehungen zwischen Klassen werden durch Verknüpfungen angezeigt. Eine Nutzungsbeziehung besteht dann, wenn als Typ einer Klassen- oder Instanzvariablen eine nicht-elementare andere Klasse angegeben wird. Weitere Verknüpfungen bestehen zwischen Methoden, die innerhalb einer Vererbungshierarchie redefiniert werden.

Klassenkategorien: Ähnlich wie schon bei den Methodenkategorien verallgemeinert das KHS den Ansatz, der von VisualWorks vorgegeben wird. Klassenkategorien, die durch spezielle Strukturknoten repräsentiert werden, können beliebig geschachtelt werden. Klassen können Teil mehrerer Kategorien sein. Damit ist eine erheblich differenziertere Modellierung des Definitions- und Verwendungszusammenhangs einzelner Klassen möglich.

Werkzeuge: Es soll hier nicht nochmals aufgeführt werden, wie die bereits bekannten Werkzeuge des KHS — graphische Übersichten, Suchwerkzeuge — zur Verbesserung der Interaktion gegenüber der Smalltalk-Entwicklungsumgebung beitragen, die über ein derartiges Instrumentarium nicht verfügt. Hier soll nur kurz auf einen Aspekt kon-

textorientierter Suche hingewiesen werden, der für den Smalltalk-Entwickler besonders relevant ist. Eine während der Wartung von Software häufig genutzte Aufgabe ist es, festzustellen, wo überall in der Software eine bestimmte Methode genutzt wird. Dies wird bei sprechenden Methodennamen ('size', 'first'), die innerhalb der Software häufiger auftreten, erheblich erschwert. Hier kann durch Einschränkung des Suchraums auf relevante Teile der Software die Treffergenauigkeit erheblich erhöht werden.

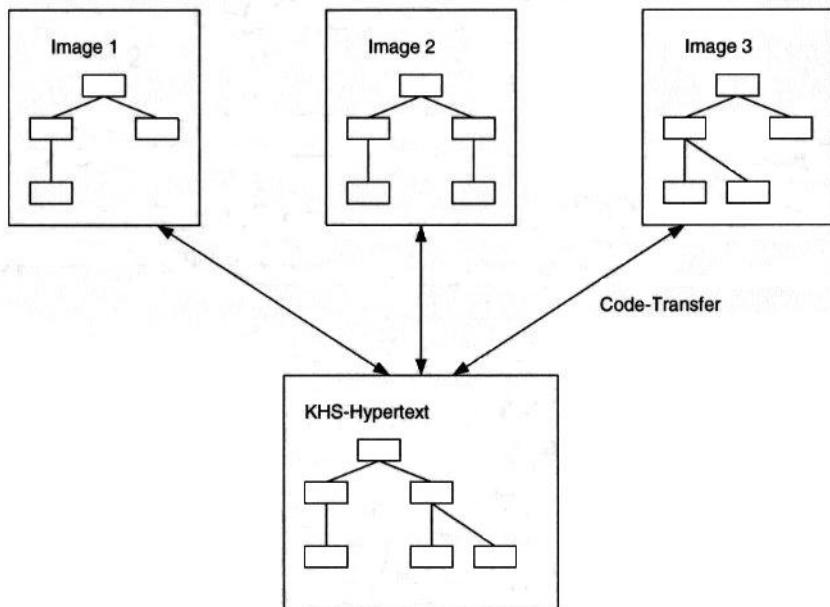


Abbildung 23 Versionskonflikte zwischen verschiedenen Smalltalk-Images und dem Software-Hypertext

4.2.4 Gemeinsamer Zugriff auf Software

Die kooperative Erstellung von Software kann, es wurde bereits darauf hingewiesen, als ein Spezialfall der computerunterstützten Gruppenarbeit angesehen werden. Insofern können alle Überlegungen zur Kommunikation in Gruppen via e-mail und zur Nutzung gemeinsamer Informationsräume auf die Software-Entwicklung übertragen werden. Auch Software-Segmente werden in Projekten entwickelt, denen Arbeitsbereiche mit restriktierten Zugriffsrechten innerhalb des gesamten Hypertexts zugewiesen werden. Auch hier ist eine Arbeitsteilung vom Entwurf über Entwicklung und Codierung bis zur Qualitätssicherung möglich, die durch Workflows unterstützt werden kann. Diese Aspekte sollen hier nicht wieder aufgerollt werden. Vielmehr soll hier nur die Fragestellung der Schnittstelle zwischen einem arbeitsteilig genutzten Informationssystem, nämlich dem KHS, und einer individuell zu nutzenden Entwicklungsumgebung für Smalltalk angesprochen werden.

Zunächst ist festzustellen, daß die Integration von Hypertextsystem und Entwicklungs-umgebung ohne Eingriff in das Laufzeitsystem erfolgt. Die Verwaltung des Objektcodes, die Ausführung von Programmen und die Fehlersuche geschehen also nach wie vor außerhalb des Hypertextsystems. Des weiteren blieb auch die standardmäßige Verwaltung des Quellcodes unangetastet, so daß folgende Situationen entstehen können (s.a. Abb. 23):

- Programme werden mit den Werkzeugen des KHS modifiziert. Das lokale Image und der Hypertext spiegeln die gleiche Fassung der Software wider. Andere Projektmitarbeiter sehen die neue Version im Hypertext, müssen aber ihre Images angleichen.
- Programme werden, z.B. zu Testzwecken ohne Nutzung des KHS modifiziert. Bei Erfolg müssen die Änderungen in den Hypertext übertragen werden.

Ein Teil dieser Konsistenzprobleme ließe sich durch eine Erweiterung des Hypertext-Ansatzes auf den Objektcode vermeiden, dennoch ist immer davon auszugehen, daß nicht alle Tests und Experimente in einer vollständig geteilten Entwicklungsumgebung stattfinden können. Eine wichtige Eigenschaft der KHS-Softwarehypertexte ist deshalb, daß Instrumente zur Entdeckung und Qualifizierung von Versionskonflikten zur Verfügung stehen (s.a. Abb. 24). Konfliktträchtige Softwaresegmente werden gekennzeichnet und sind bis zur Behebung des Konflikts nicht mehr durch normale Editieroperationen modifizierbar. Eine Bereinigung kann durch Übertragung von Software vom Image in den Hypertext oder umgekehrt stattfinden.

The screenshot shows a window titled "KHS: Transcript". The content is a conflict report on the definition of a class named "Unit". The report is timestamped at "6:01:28 pm" and includes the following text:

```

-----
6:01:28 pm
*** information ***
conflict report on definition of: Unit
instance variables defined in hypertext only
  InHypertext
    name
    indexTerms
    changed
    obsolete
    TextConstants
instance variables defined in image only
  soundDescriptor
  versions
  actVersion
  exchangeInfo
-----
```

Abbildung 24 Analysierte Codekonflikte zwischen Hypertext und Image bezüglich einer Klassendefinition

5. Zusammenfassende Anmerkungen zu den Anwendungen des KHS

Die vorangehend eingeführten Anwendungen des KHS, die sehr heterogenen Bereichen wie Wissensrepräsentation, kooperatives Arbeiten und Software-Engineering entstammen, haben gezeigt, daß das Hypertextmodell des KHS offen für die Anpassung an sehr unterschiedliche Anwendungsgebiete ist. Einige weitere kleine Anwendungen sollen hier nur kurz erwähnt werden:

- Seit einiger Zeit wird mit dem KHS die Mail zweier Listserver automatisch archiviert und im WWW zur Verfügung gestellt. Dies sind die Server *ht-link*⁴ der Fachgruppe Hypertext der Gesellschaft für Informatik und *iw-link* des Hochschulverbandes Informationswissenschaft (HI).
- Im WS 95/96 wurden erstmals Kursmaterialien zu zwei Kursen — Marc Rittberger: Qualität von Informationsdienstleistungen und Rainer Hammwöhner: Heterogene Informationssysteme — mit dem KHS erstellt und den Studierenden im WWW angeboten.
- Seit dem Frühjahr 1996 wird der "Rundbrief Information Retrieval" der Fachgruppe "Information Retrieval" der Gesellschaft für Informatik in einer elektronischen Form mit Hilfe des KHS verwaltet und herausgegeben (Redaktion: Marc Rittberger).
- Alle Materialien im Zusammenhang mit der Vorbereitung der HIM '95, der Fachtagung für Hypertext, Information Retrieval und Multimedia in Konstanz, wurden mit dem KHS verwaltet. Dies schließt die gesamte Korrespondenz, die zugehörigen

⁴ Die hier genannten Mailinglisten und Materialien sind mittlerweile nicht mehr in der hier vorgestellten Form verfügbar (Stand: 2006).

Adreßlisten und die im WWW veröffentlichten Tagungsprogramme und Begleitinformation ein.

Diese kleinen, im letzten Fall sogar für ein singuläres Ereignis konzipierten Anwendungen belegen, daß der Konzeptionsaufwand relativ gering ist. Des weiteren wird durch die Bezugnahme zwischen den verschiedenen Anwendungen deutlich, daß der objektorientierte Charakter des Hypertextmodells des KHS die Wiederverwendung und Integration verschiedener Lösungsansätze in hohem Maße unterstützt. Der Ansatz zur Gruppenarbeit baut auf den Lösungen zur Wissensrepräsentation auf und liefert seinerseits wieder Verfahren, die im Software-Engineering eingesetzt werden können.

6. Literaturangaben

- [Akscyn et al. 87] R. Akscyn, D. McCracken und E. Yoder. KMS: Distributed Hypermedia System for Managing Knowledge in Organizations. In *Hypertext '87 Papers, Chapel HUI, NC, University of North Carolina*, S. 1-20, 1987.
- [Ansellem 95] M. Ansellem. ChyPro: A hypermedia programming environment for Smalltalk-80. In W. Oethoff (Hrsg.), *ECOOP'95 Object oriented programming*, S. 449-470. Springer, 1995.
- [Bigelow & Riley 87] J. Bigelow und V. Riley. Manipulating Source Code in Dynamic Design. In *Hypertext '87 Papers, Chapel HUI, NC, University of North Carolina*, S. 397-408, 1987.
- [Bigelow 88] J. Bigelow. Hypertext and CASE. *IEEE Software*, Bd. 5, Nr. 2, S. 23-27, 1988.
- [Boldyreff 94] C. Boldyreff. Software engineering design. In D. Rosenberg und C. Hutchinson (Hrsg.), *Design issues in CSCW*, S. 139-150. Springer, 1994.
- [Bolter & Joyce 87] J.D. Bolter und M. Joyce. Hypertext and Creative Writing. In *Hypertext '87 Papers, Chapel HUI, NC, University of North Carolina*, S. 41-50, 1987.
- [Booch 91] G. Booch. *Object oriented design with applications*. Benjamin/Cummings, Redwood City, Ca, 1991.
- [Borenstein 92] N.S. Borenstein. Computational mail as network infrastructure for computer-supported cooperative work. In J. Turner und R. Kraut (Hrsg.), *CSCW '92 Sharing Perspectives, Proc. of the Conference on Computer-Supported Cooperative Work*, S. 67-74, 1992.
- [Brachman & Schmolze 85] R. Brachman und J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, Bd. 9, Nr. 2, S. 171-216, 1985.
- [Brodt 94] T. Brodt. Unterstützung des Software-Entwurfs durch Hypertext. Diplomarbeit, Universität Stuttgart, Institut für Informatik, 1994.
- [Brown 87] P.J. Brown. Turning Ideas into Products: The Guide System. In *Hypertext '87 Papers, Chapel HUI, NC, University of North Carolina*, S. 33-40, 1987.
- [Bullen & Bennett 90] C.V. Bullen und J.L. Bennett. Learning from user experience with groupware. In *Proc. of the CSCW '90*, S. 291-316, 1990.
- [Clitherow et al. 89] P. Clitherow, D. Riecken und M. Muller. VISAR: a System for inference and navigation in hypertext. In *Proc. Hypertext '89, Pittsburgh*, S. 293-304, 1989.
- [Cohen & Kjeldsen 87] P.R. Cohen und R. Kjeldsen. Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Information Processing and Management*, Bd. 23, Nr. 4, S. 255-268, 1987.

- [Collier 87] G.H. Collier. Thoth-II Hypertext with Explicit Semantics. In *Hypertext '87 Papers, Chapel Hill, NC, University of North Carolina*, S. 269-289, 1987.
- [Conklin & Begeman 87] J. Conklin und M.L. Begeman. gIBIS: A Hypertext Tool for Team Design Deliberation. In *Hypertext '87 Papers, Chapel Hill, NC, University of North Carolina*, S. 247-251, 1987.
- [Conklin 93] J. Conklin. Capturing organizational memory. In R.M. Baecker (Hrsg.), *Readings in groupware and computer-supported cooperative work - assisting human-human collaboration*, S. 561-565. Morgan Kaufmann Publishers, 1993.
- [Davenport & McKim 95] E. Davenport und G. McKim. Groupware. In M.E. Williams (Hrsg.), *Annual review of information science and technology*, S. 115-159. Information Today, 1995.
- [Diaper 95] D. Diaper. Small CSCW systems design. In K. Spurr, P. Layzell, L. Jennison und N. Richards (Hrsg.), *Computer supported cooperative work*, S. 89-107. John Wiley & Sons, 1995.
- [Dourish & Bellotti 92] P. Dourish und V. Bellotti. Awareness and coordination in shared workspaces. In J. Turner und R. Kraut (Hrsg.), *CSCW '92 Sharing Perspectives, Proc. of the Conference on Computer-Supported Cooperative Work*, S. 107-114, 1992.
- [Dugerdil 98] P. Dugerdil. Inheritance mechanisms in the OBJLOG language: multiple selective and multiple vertical with points of view. In M. Lenzerini, D. Nardi und M. Simi (Hrsg.), *Inheritance hierarchies in knowledge representation and programming languages*, S. 245-256. John Wiley & Sons, 1998.
- [Evans 90] R. Evans. Expert Systems and Hypercard. *Byte*, Bd. 15, Nr. 1, S. 317-324, 1990.
- [Fischer & McCall 89] G. Fischer und McCall. JANUS: Integrating Hypertext with a Knowledge-Based Design Environment. In *Proc. Hypertext '89, Pittsburgh*, S. 1 OS-117. ACM, 1989.
- [Flores et al. 88] F. Flores, M. Graves, B. Hartfield und T. Winograd. Computer Systems and the design of organizational interaction. *ACM Transactions on Office Information Systems*, Bd. 6, Nr. 2, S. 153-172, 1988.
- [Frost 86] R. Frost. *Introduction to Knowledgebase Systems*. Macmillan Publishing Company, New York, 1986.
- [Garbe 95] G. Garbe. *Multimediasysteme im naturwissenschaftlichen Lehren und Lernen. Evaluation zu Akzeptanz, Nutzungsverhalten und Lerneffizienz*. Dissertation, Universität Konstanz, Informationswissenschaft, 1995.
- [Garg & Scacchi 87] P.K. Garg und W. Scacchi. On Designing Intelligent Hypertext Systems for Information Management in Software Engineering. In *Hypertext '87 Papers, Chapel Hill NC, University of North Carolina*, S. 409-432, 1987.
- [Gloor 90] P.A. Gloor. *Hypermedia-Anwendungsentwicklung*. B.G. Teubner, Stuttgart, 1990.
- [Goldberg et al. 92] Y. Goldberg, M. Safran und E. Shapiro. Active mail: a framework for implementing groupware. In J. Turner und R. Kraut (Hrsg.), *CSCW '92 Sharing Perspectives, Proc. of the Conference on Computer-Supported Cooperative Work*, S. 75-83, 1992.
- [Goldberg 84] A. Goldberg. Smalltalk-80 The Interactive programming environment. Addison-Wesley, Reading, 1984.

- [Greif & Sarin 87] I. Greif und S. Sarin. Data sharing in group work. *ACM Transactions on Information Systems*, Bd. 5, Nr. 2, S. 187-211, 1987.
- [Haake et al. 94] J.M. Haake, N.A. Streitz und C.M. Neuwirth. Coexistence and transformation of informal and formal structures: Requirements for more flexible hypermedia Systems. In Proc. ECHT '94, Edinburgh, S. 1-12, 1994.
- [Halasz 88] F.G. Halasz. Reflections on NoteCards: Seven issues for the next generation of hypermedia Systems. *Communications of the ACM*, Bd. 31, Nr. 7, S. 836-852, 1988.
- [Hammwöhner & Kuhlen 94] R. Hammwöhner und R. Kuhlen. Semantic control of open hypertext Systems by typed objects. *Journal of Information Science*, Bd. 20, Nr. 3, S. 193-202, 1994.
- [Hammwöhner & Rittberger 93] R. Hammwöhner und M. Rittberger. KHS - ein offenes Hypertext-System. In G. Knorz, J. Krause und C. Womser-Hacker (Hrsg.), *Information Retrieval '93. Von der Modellierung zur Anwendung. Proceedings der Tagung Information Retrieval '93*, S. 208-222, 1993.
- [Hofmann & Simon 95] M. Hofmann und L. Simon. Problemlösung Hypertext. Hanser, München Wien, 1995.
- [Jablonski 95] S. Jablonski. Workflow-Management-Systeme: Motivation, Modellierung, Architektur. Informatik Spektrum, Bd. 18, Nr. 1, S. 13-24, 1995.
- [Jarwa & Bruandet 90] S. Jarwa und M.-F. Bruandet. A hypertext database model for information management in Software engineering. In A.M. Tjoa und R. Wagner (Hrsg.), Proc. of the Int. Conf. on Database and Expert Systems (DEXA '90), S. 69-75, 1990.
- [Jarwa & Bruandet 92] S. Jarwa und M.-F. Bruandet. Elen prototype: an active hypertext system for document management in Software engineering. In A.M. Tjoa und L Ramos (Hrsg.), Proc. of the Int. Conf. on Database and Expert Systems (DEXA '92), S. 450-456, 1992.
- [Jonassen 89] D. Jonassen. Semantic Networking Approaches to Structuring Hypertext. In Proc. Hypertext II, York, 1989.
- [Jordan et al. 89] D.S. Jordan, D.M. Rüssel, A.-M. S. Jensen und R. A. Rogers. Facilitating the Development of Representations in Hypertext with IDE. In Proc. Hypertext '89, Pittsburgh, S. 93-104, 1989.
- [Kaindl & Snaprud 91] H. Kaindl und M. Snaprud. Hypertext and structured object representation: a unifying view. In Proc. Hypertext' 91, SanAntonio, S. 345-358, 1991.
- [Kaindl 93] H. Kaindl. The missing link in requirements engineering. ACM SIGSOFT Software Engineering notes, Bd. 18, Nr. 2, S. 30-39, 1993.
- [Karlsson & Morel 95] E.-A. Karlsson und J.-M. Morel. The impact of reuse on software-quality. In P. Nesi (Hrsg.), *Objective Software quality*, S. 124-137. Springer, 1995.
- [Knopik & Ryser 89] T. Knopik und S. Ryser. AI-Methods for Structuring Hypertext-Information. In *Proc. Hypertext H, York*, 1989.
- [Knuth 84] D.E. Knuth. Literate programming. *The Computer Journal*, Bd. 27, Nr. 2, S. 97-111, 1984.
- [Koh & Chua 89] T.T. Koh und T.S. Chua. On the Design of a Frame-Based Hyper-media System. In *Proc. Hypertext II, York*, 1989.
- [Kuhlen 91] R. Kuhlen. *Hypertext — ein nicht-lineares Medium zwischen Buch und Wissensbank*. Springer, Heidelberg, New York, 1991.

- [Kuhlen 94] R. Kuhlen. Elektronische nicht-lineare Dossiers in offenen Hypertextsystemen. In W. Rauch, F. Strohmeier, H. Hiller und C. Schlägl (Hrsg.), *Proc. ISI'94*, Graz, S. 303-310, 1994.
- [Kuhlen 95] R. Kuhlen. *Informationsmarkt. Chancen und Risiken der Kommerzialisierung von Wissen*. Universitätsverlag Konstanz, Konstanz, 1995.
- [Lai et al. 88] K.-Y. Lai, T.W. Malone und K.-C. Yu. Object Lens: A 'spreadsheet' for cooperative work. *ACM Transactions on Office Information Systems*, Bd. 6, Nr. 4, S. 332-353, 1988.
- [Lalonde & Pugh 90] W.R. Lalonde und J.R. Pugh. *Inside Smalltalk*. Prentice-Hall International, London, 1990.
- [Landow 92] G.P. Landow. *Hypertext. The Convergence of Contemporary Critical Theory and Technology*. The Johns Hopkins University Press, Baltimore, 1992.
- [Lenzerini et al. 89] M. Lenzerini, D. Nardi und M. Simi. *Inheritance hierarchies in knowledge representation and programming languages*. John Wiley & Sons, Chichester etc., 1989.
- [Lutz et al. 90] E. Lutz, H.v. Kleist-Retzow und K. Hoerning. MAFIA - An active mail-filter-agent for an intelligent document processing support. *ACM SIGOIS Bulletin*, Bd. 11, Nr. 4, S. 16-32, 1990.
- [Mackay 88] W.E. Mackay. Diversity in the use of electronic mail: a preliminary inquiry. *ACM Transactions on Office Information Systems*, Bd. 6, Nr. 4, S. 380-397, 1988.
- [Malone & Crowston 90] T.W. Malone und K. Crowston. What is coordination theory and how can it help design cooperative work. In *Proc. of the CSCW'90*, S. 357-370, 1990.
- [Malone et al. 86] T.W. Malone, K.R. Grant und F. Turbak. The Information Lens: an intelligent System for information sharing in organizations. In M. Mantei und P. Orbeton (Hrsg.), *Human factors in computing Systems. Proc. of the CHI'86 Conf.*, S. 1-8, 1986.
- [Malone et al. 87a] T.W. Malone, K.R. Grant, K. Lai, R. Rao und D. Rosenblitt. Semi-structured messages are surprisingly useful for computer-supported coordination. In *Computer Supported Cooperative Work Conference*, 1987.
- [Malone et al. 87b] T.W. Malone, K.R. Grant, F. Turbak, S. Brobst und M. Cohen. Intelligent information-sharing Systems. *Communications of the ACM*, Bd. 30, Nr. 5, S. 390-402, 1987.
- [Malone et al. 92] T.W. Malone, K.Y. Lai und C. Fry. A radically tailorable tool for cooperative work. In *Proc. CSCW'92, Toronto, Canada*, S. 289-297, 1992.
- [Marshall & Rogers 92] C.C. Marshall und R.A. Rogers. Two years before the mist: experiences with Aquanet. In *Proc. ECHT'92, Milano*, S. 53-62, New York, 1992. ACM.
- [Marshall & Shipman 93] C.C. Marshall und F.M. Shipman. Searching for the missing link: discovering implicit structure in spatial hypertext. In *Proc. Hypertext' 93, Seattle*, S. 217-230, New York, 1993. ACM.
- [Marshall et al. 91] C.C. Marshall, F.G. Halasz, R.A. Rogers und W.C. Jannssen. Aquanet: a hypertext tool to hold your knowledge in place. In *Proc. Hypertext' 97, San Antonio*, S. 261-275, New York, 1991. ACM.
- [Marshall et al. 94] C.C. Marshall, F.M. Shipman und J.H. Coombs. VIKI: Spatial hypertext supporting emergent structure. In *Proc. ECHT '94, Edinburgh*, S. 13-23, New York, 1994. ACM.

- [McAleese 90] R. McAleese. Concepts as Hypertext Nodes: The Ability to Navigate through Hypertext Nets. In *Proc. NATO Advanced Research Workshop "Designing Hypertext/Hypermedia for Learning"*, S. 97-115, 1990.
- [Mössenböck 90] H. Mössenböck. Ein Programmeditor mit Hypertext-Fähigkeiten. In *Hypertext und Hypermedia*, S. 43-52, Berlin, Heidelberg, New York, 1990. Springer.
- [Musliner et al. 92] D.J. Musliner, J.W. Dolter und K.G. Shin. BibDb: A bibliographic database for collaboration. In J. Turner und R. Kraut (Hrsg.), *CSCW '92 Sharing Perspectives, Proc. of the Conference on Computer-Supported Cooperative Work*, S. 386-393, 1992.
- [Nanard & Nanard 91] J. Nanard und M. Nanard. Using structured types to incorporate knowledge into hypertext. In *Proc. Hypertext' 91, San Antonio*, S. 329-343, New York, 1991. ACM.
- [Nelson 92] T.H. Nelson. *Literary Machines 93.1*. Mindful Press, Sausalito CA, 1992.
- [Neubert & Oberweis 92] S. Neubert und A. Oberweis. Einsatzmöglichkeiten von Hypertext bei Software Engineering and Knowledge Engineering. In *Hypertext und Hypermedia* 1992, S. 162-174, Berlin, Heidelberg, New York, 1992. Springer.
- [Neuwirth et al. 90] C.M. Neuwirth, R. Kaufer, D.S. Chandhok und J. Morris. Issues in the design of computer support for co-authoring and commenting. In *Proc. of the CSCW '90*, S. 183-195, 1990.
- [Nicolson & Thomlinson 91] R.L Nicolson und P. Thomlinson. USHIR: A knowledge-based hypermedia System. *Hypermedia*, Bd. 3, Nr. 1, S. 1-33, 1991.
- [Nicolson 89] R.L Nicolson. Towards the third generation: the case for IKBH (Intelligent Knowlege Based Hypermedia environments. In *Proc. Hypertext 77, York*, 1989.
- [Nielsen 93] J. Nielsen. *Hypertext & Hypermedia*. Academic Press Professional, Boston etc., 1993.
- [Ortner 93] E. Ortner. KASPER Konstanzer Sprachkritik-Programm für das Software-Engineering. Bericht Bericht 36-93, Universität Konstanz, Informationswissenschaft, 1993.
- [Osterbye 93] K. Osterbye. Literate Smalltalk Programming. Bericht R 93-2025, Aalborg University, Institute for Electronic Systems, Dep. of Mathematics and Computer Science, 1993.
- [Palme 95] J. Palme. *Electronic mail*. Artech House, Boston, London, 1995.
- [Pollock 88] S. Pollock. A rule-based message filtering System. *ACM Transactions on Office Information Systems*, Bd. 6, Nr. 3, S. 232-254, 1988.
- [Postel et al. 88] J.B. Postel, G.G. Finn, A.R. Katz und J.B. Reynolds. An experimental multi-media mail system. *ACM Transactions on Office Information Systems*, Bd. 6, Nr. 1, S. 63-81, 1988.
- [Puppe 91] F. Puppe. Wissensrepräsentation mit Regeln. In P. Stuß(Hrsg.), *Wissensrepräsentation*, S. 123-130. Oldenbourg, 1991.
- [Puppe 93] F. Puppe. *Systematic introduction to expert systems - Knowledge representations and problem-solving methods*. Springer, Heidelberg etc., 1993. ^
- [Rada 91] R. Rada. *Hypertext: from text to expertext*. McGraw-Hill, 1991.

- [Reimer 89] U. Reimer. *FRM: Ein Frame-Repräsentationsmodell und seine formale Semantik. Zur Integration von Datenbank- und Wissensrepräsentationsansätzen*. Springer, Heidelberg etc., 1989.
- [Reimer 91] U. Reimer. *Einführung in die Wissensrepräsentation - netzartige und schemabasierte Repräsentationsformate*. Teubner, Stuttgart, 1991.
- [Reinhardt 93] A. Reinhardt. Smarter e-mail is coming. *Byte*, S. 90-108, 1993.
- [Richartz & Rüdebusch 90] M. Richartz und T.D. Rüdebusch. Collaboration in hypermedia environments. In D. Jonassen (Hrsg.), *Proc. NATO Advanced Research Workshop "Designing Hypertext/Hypermedia for Learning"*, S. 311-317, 1990.
- [Rüssel 90] D. Rüssel. Hypermedia and representation. In N.A. Streitz und P.A. Gloor (Hrsg.), *Hypertext und Hypermedia*, S. 1-9, 1990.
- [Sametinger & Stritzinger 91] J. Sametinger und A. Stritzinger. Ein Hypertext-Editor zur Software-Wartung. In H. Maurer (Hrsg.), *Hypertext/Hypermedia '91 Tagung der GI, SI und ÖCG*, S. 249-256, 1991.
- [Schatz 93] B.R. Schatz. Building an electronic community System. In R.M. Baecker (Hrsg.), *Readings in groupware and computer-supported cooperative work - assisting human-human collaboration*, S. 550-560. Morgan Kaufmann Publishers, 1993.
- [Schienmann 96] B. Schienmann. *Objektorientierter Fachentwurf. Ein terminologiebasierter Ansatz für die Konstruktion von Anwendungssystemen*. Dissertation, Universität Konstanz, Informationswissenschaft, 1996.
- [Schönthaler & Nemeth 90] F. Schönthaler und T. Nemeth. *Software-Entwicklungs-werkzeuge: Methodische Grundlagen*. B.G. Teubner, Stuttgart, 1990.
- [Sedes 92] F. Sedes. A hypertext information system for reusable software component retrieval. In A.M. Tjoa und L Ramos (Hrsg.), *Proc. of the Int. Conf. on Database and Expert Systems (DEKA '92)*, S. 457-462, 1992.
- [Shepherd et al. 90] A. Shepherd, N. Mayr und A. Kuchinsky. Strudel - an extensible electronic conversation toolkit. In *Proc. of the CSCW '90*, S. 93-104, 1990.
- [Shipman & McCall 94] F.M. Shipman und R. McCall. Supporting knowledge-base evolution with incremental formalization. In *Proc. CHI'94, Boston, Mass.*, S. 285-291, 1994.
- [Shipman et al. 89] F.M. Shipman, RJ. Chaney und G.A. Gorry. Distributed hypertext for collaborative research: The Virtual Notebook System. In *Proc. Hypertext '89, Pittsburgh*, S. 129-135, 1989.
- [Smith et al. 87] J.B. Smith, S.F. Weiss und G.J. Ferguson. A hypertext writing environment and its cognitive basis. In *Hypertext '87 Papers, Chapel HUI, NC, University of North Carolina*, S. 195-224, 1987.
- [Snaprud & Kaindl 92] M. Snaprud und H. Kaindl. Knowledge acquisition using hypertext. *Expert Systems with Applications*, Bd. 5, Nr. 3-4, S. 369-375, 1992.
- [Snaprud & Kaindl 94] M. Snaprud und H. Kaindl. Types and inheritance in hypertext. *Int. J. Human-Computer Studies*, Bd. 41, Nr. 1/2, S. 223-241, 1994.
- [Spribille 95] P. Spribille. E-mail als elektronische Post und wissenschaftliche Kommunikation. Vergleich von SunOS-E-mail-Tool und KHS-E-Mail-Tool hinsichtlich der Integration informationeller Mehrwerte. Diplomarbeit, Universität Konstanz - Informationswissenschaft, 1995.

- [Streitz et al. 92] N.A. Streitz, J.M. Haake, J. Hannemann, A. Lemke, W. Schuler, H. Schutt und M. Thüring. SEPIA: A cooperative hypermedia authoring environment. In *Proc. ECHT '92, Milano*, S. 11-22, New York, 1992. ACM.
- [Streitz 90] N.A. Streitz. Hypertext; Ein innovatives Medium zur Kommunikation von Wissen. In N.A. Streitz und P.A. Gloor (Hrsg.), *Hypertext und Hypermedia*, S. 10-27, 1990.
- [Teufel et al. 95] S. Teufel, C. Sauter, T. Mühlherr und Bauknecht. K. *Computerunterstützung bei der Gruppenarbeit*. Addison-Wesley, Bonn etc., 1995.
- [Tochtermann & Zink 95] K. Tochtermann und V. Zink. Artificial intelligence and hypermedia. In R. Rada und K. Tochtermann (Hrsg.), *Expertmedia - expert systems and hypermedia*, S. 61-89. World Scientific, 1995.
- [Volst 94] A. Volst. Information ist der Mehrwert von Netzwerken in den Wissenschaften or The benefit of computer-networks in science. In W. Rauch, F. Strohmeier, H. Hiller und C. Schlägl (Hrsg.), *Proc. ISI '94*, Graz, S. 559-571, 1994.
- [Weyer & Borning 85] S.A. Weyer und A.H. Borning. A Prototype Electronic Encyclo-pedia. *ACM Transactions on Office Information Systems*, Bd. 3, Nr. 1, S. 63-88, 1985.
- [Winograd & Flores 86] T. Winograd und F. Flores. *Understanding Computers and cognition*. Ablex Publishing Corporation, Norwood, NJ, 1986.
- [Yankelovich et al. 88] N. Yankelovich, J. Haan B, N.K. Meyrowitz und S.M. Drucker. Intermedia: The concept and the construction of a seamless information environment. *IEEE Computer*, Bd. 21, Nr. 1, S. 81-96, 1988.
- [Yetim 92] F. Yetim. Ein Ansatz zur flexiblen Gestaltung benutzerobergereehter Antworten für Erklärungsdialo. In H.H. Zimmermann et al. (ed.): *Mensch und Maschine - Informationelle Schnittstellen der Kommunikation. Proceedings des 3. Internationalen Symposiums für Informationswissenschaft (ISI '92)*, S. 338-353, Konstanz, 1992. Universitätsverlag Konstanz.
- [Yetim 94] F. Yetim. *Erklärungen im Kontext der Mensch-Computer-Interaktion. Ein Konzept zur Integration der Methoden von Hypertext- und Künstlichen Intelligenzforschung*. Dissertation, Universität Konstanz, Fakultät für Verwaltungswissenschaft, 1994.
- [Zink 95] V. Zink. Die Wissensbasis im Konstanzer Hypertext System (KHS). Bericht 77-95, Universität Konstanz - Informationswissenschaft, 1995.