Dreidimensionale Visualisierung von Wissen unter Berücksichtigung von Metadaten und Raumkonzepten

Birgit Labusch

Diplomarbeit
August 2001

Fachbereich Informatik
Universität Hamburg

Erstbetreuer: Prof. Dr. Horst Oberquelle
Zweitbetreuer: Dr. Ralf Röhrig

Inhaltsverzeichnis

1 Einleitung **7**

2 Wissensmanagement **11**
2.1 Zeichen – Daten – Information – Wissen 11
2.2 Warum Wissensmanagement? 14
2.3 Bausteine des Wissensmanagements 17
2.4 Transparenz durch Wissenskarten 24

3 Evaluationsmethoden **31**
3.1 Analytische Evaluation 32
3.2 Evaluation durch Experten 32
3.3 Evaluation durch Beobachtung 36
3.4 Evaluation durch Befragung 38
3.5 Experimentelle Evaluation 39

4 Wissensmanagement-Systeme **41**
4.1 SERPersonalBrain 42
4.1.1 Retrieve 42
4.1.2 Classify 49
4.2 Inxight Tree Studio 59
4.2.1 Inxight Tree Studio (Hauptprogramm) 59
4.2.2 Star Tree Designer 63
4.3 Mind Manager 68

5 Metadaten und Wissensmanagement **73**
5.1 Beschreibungssprachen für Metadaten 74
5.1.1 Dublin Core 74
5.1.2 SGML 77
5.1.3 HTML 81
5.1.4 XML 84

5.1.5 RDF 89
5.2 „Versteckte“ Metadaten 94

6 Visualisierung von Informationen 103
6.1 Datenstrukturen für die Informations-Visualisierung 103
6.1.1 Eindimensionale Datenstrukturen 104
6.1.2 Zweidimensionale Datenstrukturen 106
6.1.3 Dreidimensionale Datenstrukturen 107
6.1.4 Multi-dimensionale Datenstrukturen 108
6.1.5 Temporale Datenstrukturen 112
6.1.6 Baumstrukturen 113
6.1.7 Netzwerkstrukturen 117
6.1.8 Metaphern im Informationsraum 118
6.2 Gründe für die Verwendung von drei Dimensionen 120
6.3 Layoutgestaltung 121

7 Metadaten und Visualisierung 125
7.1 Der Algorithmus von Card, Mackinlay und Shneiderman 125
7.2 Visualisierung mit Hilfe von Raumkonzepten 129

8 Wissensmanagement in der Praxis 138
8.1 Das Entity/Relationship-Modell 138
8.2 Die KnowledgeBase von Lufthansa Systems AS 140
8.3 Das Intranet der Fernseh AG 150

9 Benutzungstests 155
9.1 Benutzungstests mit dem Mock-up 155
9.2 Die VRML-Prototypen 163
9.3 Benutzungstests mit dem VRML-Prototypen 171

10 Ausblick 177

Literaturverzeichnis 179

A Videoprotokolle 186

B Farbtafeln 209

Danksagung

Mein besonderer Dank gilt Herrn Ralf Röhring von der Firma `Lufthansa Systems AS` sowie Frau Marianne Steenken von der `Fernseh AG`[1], die es mir erlaubten, die Wissensbasen ihrer Firmen im Rahmen der nachfolgenden Arbeit verwenden zu dürfen.

[1] Der Name der Firma wurde auf Wunsch der Verantwortlichen geändert.

Kapitel 1

Einleitung

Viele Firmen besitzen ein noch zu wenig genutztes Kapital: das Wissen ihrer Mitarbeiter. Um in der Zukunft im immer härter werdenden Wettbewerb bestehen zu können, muss diese Ressource optimal genutzt werden. Dazu gehört nicht nur, dass das Wissen vorhanden, sondern auch, dass es auffindbar ist. Eine gute Visualisierung kann hierbei eine erhebliche Unterstützung bieten.
Während das Wissensmanagement Konzepte und Methoden liefert, um das Wissen zu vermehren, zu verteilen und für die Lösung vielfältiger Aufgaben eines Unternehmens zu nutzen, helfen Metadaten bei der Ablage und beim Wiederfinden der Informationen aus Datenbanken oder Dokumentarchiven.
Für die Präsentation von Wissen gibt es eine Vielzahl von Visualisierungsmöglichkeiten. Das Auffinden der geeigneten Präsentationsform stellt sich dabei häufig als schwierig und zeitaufwendig heraus. Eine Unterstützung können hierbei Metadaten bieten, die auf Raumkonzepte abgebildet werden, welche dann ihrerseits das Auffinden einer geeigneten Visualisierungsform erleichtern können.

Motivation und Ziele der Arbeit

Durch die Mitarbeit an einem Projekt bei der `DaimlerChrysler Aerospace AG (Dasa)` machte ich mit dem interessanten Thema Wissensmanagement meine erste Bekanntschaft. Dort nutzte man sog. *Mind Maps* (siehe hierzu Abschnitt 4.3) zur Visualisierung des Wissensbestandes. Diese Karten waren so groß und unhandlich, dass sie weder auf eine Bildschirmseite, noch auf einen Papierausdruck der Größe A4 passten (vorausgesetzt sie sollten in einer lesbaren Größe dargestellt werden). So kam ich auf die Idee, dass es vielleicht andere Visualisierungsmöglichkeiten geben könnte, die dieses Manko nicht aufweisen und zusätzlich eine Erleichterung des Auffindens von Wissen ermöglichen könnten. Bereits zu diesem

Zeitpunkt schwebte mir die Idee vor, dass ein dreidimensionaler Lösungsansatz vielversprechend sein könnte.
Die Idee einer Diplomarbeit war also geboren. Was jetzt noch fehlte war eine Wissensbasis, anhand derer ich die Idee der dreidimensionalen Visualisierung einer Wissensbasis umsetzen konnte. Aus den Erfahrungen, die ich bei der `Dasa` gesammelt hatte, wusste ich, wie aufwendig es ist, eine Wissensbasis zu erstellen und wollte aus diesem Grund gerne auf eine bereits bestehende zurückgreifen.
Zu der benötigten Wissensbasis verhalfen mir meine beiden Betreuer. Meinem Erstbetreuer – Horst Oberquelle – habe ich den Kontakt zu meinem Zweitbetreuer – Ralf Röhrig – zu verdanken, denn er war es, der mich auf ein Treffen des `HIForums`[1] aufmerksam machte, in dem das Thema Wissensmanagement diskutiert werden sollte und mir riet, auch an diesem Treffen teilzunehmen. Dort lernte ich meinen Zweitbetreuer kennen, einen der Hauptverantwortlichen für das Wissensmanagement bei `Lufthansa Systems AS`. Er war es dann auch, der mir die *Knowledge Base* des Unternehmens für meine Forschungszwecke zur Verfügung stellte. Außerdem brachte er die Idee in meine Arbeit ein, Metadaten und Raumkonzepte zu betrachten und zu untersuchen, inwieweit sie auf Visualisierungskonzepte angewendet werden können.
Um nicht nur auf eine Wissensbasis zurückgreifen zu müssen, sollten ursprünglich die Webseiten des Fachbereichs Informatik der Universität Hamburg als Vergleichs-Wissensbasis dienen. Im Verlauf der Arbeit stellte sich aber heraus, dass diese für meine Arbeit eher ungeeignet waren. Als sich mir dann die Möglichkeit bot, die Wissensbasis der `Fernseh AG` – die zu Vergleichszwecken geeigneter schien, da sich hier ähnliche Strukturen wie bei `Lufthansa Systems AS` wiederfinden ließen – in die Arbeit mit aufzunehmen, wurde in Absprache mit meinem Erstbetreuer diese Wissensbasis statt der Fachbereichsseiten in meiner Arbeit verwendet.
Mit Hilfe dieser Wissensbasen war ich in der Lage, meine dreidimensionalen Prototypen zu entwickeln und nach software-ergonomischen Kriterien zu untersuchen. Tests mit potentiellen Nutzern sollten zeigen, wie gut das von mir entworfene User-Interface gelungen war, ob eine effiziente Arbeit mit ihnen möglich sein würde und wo es noch Schwierigkeiten gab.

Aufbau der Arbeit

Nachdem dieses Kapitel die Ziele der Arbeit aufzeigen sollte, wird der Leser im zweiten Kapitel in das Gebiet des Wissensmanagements eingeführt. Hier werden

[1] HIForum e.V. Hamburger Informatik-Forum

Gründe für den Einsatz von Wissensmanagement geliefert und erklärt, warum die Visualisierung von Wissensbeständen anhand von Wissenskarten so wichtig ist.
Um mir einen Überblick über bereits bestehende Wissensmanagementsysteme zu verschaffen, werde ich in Kapitel 4 drei Anwendungen aus dem Bereich des Wissensmanagements mit der Methode der Heuristischen Evaluation untersuchen. Das Kapitel soll auch aufzeigen, was heutige Wissensmanagement-Systeme dem Benutzer bieten können und auf welchem Stand der Technik – in Bezug auf die Wissensvisualisierung – sich diese Anwendungen befinden.
Zuvor werde ich im dritten Kapitel einen allgemeinen Überblick über Evaluationsmethoden geben, da in meiner Arbeit verschiedene Evaluationsmethoden Verwendung gefunden haben. Auf diese Weise soll es dem Leser erleichtert werden nachzuvollziehen, warum ich welche Methode verwendet habe.
Anschließend liefere ich eine Einführung in die Metadaten. Neben gebräuchlichen Beschreibungssprachen für Metadaten zeige ich, dass auch in Datenstrukturen Metadaten enthalten sind.
Bei der Visualisierung muss man die Dimension der Daten(-struktur) und die Dimension der Visualisierungsform unterscheiden. So ist z.B. Text – eine Aneinanderreihung von Zeichen – eine eindimensionale Datenstruktur (eindimensionales Array oder lineare Liste). Ein Buch – ganz gleich ob real oder virtuell – nutzt mit der Visualisierungsform „Seite“ zwei Dimensionen zur Darstellung. Das Buch ist also eine zweidimensionale Visualisierungsform. Man kann aber auch eine dreidimensionale Visualisierungsform zur Textdarstellung nehmen (z.B. WebBook). Das Kapitel 6 zeigt ausgehend von der Eigenschaft der Datenstruktur unterschiedliche Visualisierungsformen auf.
Kapitel 7 verknüpft die beiden vorangegangen Kapitel. Es stellt zwei Algorithmen vor, wie Metadaten zur Auffindung einer geeigneten Visualisierungsform genutzt werden können. In Abschnitt 7.2 wird dabei auch auf die Raumkonzepte eingegangen.
Wissensmanagement in der Praxis ist das Thema von Kapitel 8. Es stellt die Wissensmanagement-Lösungen bei `Lufthansa Systems AS` und der `Fernseh AG` vor. Es liefert auch die Begründung über die Entscheidung darüber, warum welche Teile für die Erstellung der Prototypen Verwendung fanden.
Die Prototypen, ein dreidimensionales Mock-up und eine der beiden Realisierungen mit VRML, wurden von potentiellen Anwendern getestet. Die Testabläufe sowie die daraus gewonnenen Erkenntnisse schildert das Kapitel 9.
Im letzten Kapitel liefere ich einen Ausblick auf mögliche weiterführende Fragestellungen.

Meine Arbeit enthält viele Abbildungen, die häufig auch farbig sind. Da es aber

zu kostspielig gewesen wäre, die ganze Arbeit auf einem Farblaserdrucker auszudrucken, habe ich mich dazu entschlossen, die Abbildungen, die für die Arbeit von großer Bedeutung bzw. in schwarz-weiß schlecht erkennbar wären, extra in einer Farbtafel aufzuführen. Wer an den restlichen Farbabbildungen Interesse hat, kann diese in der PDF-Version meiner Arbeit, die auf der beigefügten CD-ROM zu finden ist, nachschlagen.
Die CD-ROM enthält neben den Demo-Versionen der in Kapitel 4 untersuchten Anwendungen auch die VRML-Prototypen sowie ein Video aus den Benutzungstests. Die benötigte Software zur Betrachtung der VRML-Szenen oder des Videos ist enthalten.

Es sollte noch kurz erwähnt werden, dass die Arbeit nach den neuen amtlichen Rechtschreibregeln verfasst ist. Sollte ich versehentlich doch einmal die alten Regeln verwenden, so bitte ich darum, dies zu entschuldigen.

Kapitel 2

Wissensmanagement

Dieses Kapitel soll eine kurze Einführung in das Gebiet des Wissensmanagements liefern. Zuerst soll geklärt werden, in welchen Zusammenhängen der Begriff des Wissens betrachtet werden sollte. Anschließend folgt ein kurzer Überblick zur Fragestellung, warum Wissensmanagement in Unternehmen eingesetzt werden sollte gefolgt von den *Bausteinen des Wissensmanagements*, die eine Hilfestellung zur erfolgreichen Durchführung von Wissensmanagement-Projekten in Unternehmen liefern können. Zum Abschluss wird einer dieser Bausteine – die Wissensidentifikation – genauer betrachtet. Es soll beleuchtet werden, wie Wissenskarten die Wissensidentifikation unterstützen können.

2.1 Zeichen – Daten – Information – Wissen

In der Informatik spielt der Begriff der Information eine wichtige Rolle. Erst durch die Vernetzung von Informationen und einem Vorgang des Erkennes kann daraus Wissen entstehen. So wie Informationen die Grundbausteine des Wissens sind, so benötigen Informationen Daten und Daten Zeichen als Grundbausteine. Die Begriffe Zeichen, Daten, Information und Wissen bilden eine Hierarchie, wie sie in Abbildung 2.1 zu sehen ist (nach [Probst et al. 1999, Seite 36]).
Zeichen sind zusammenhangslose Datenelemente, die aus Buchstaben, Ziffern, Sonderzeichen oder ähnlichen Symbolen bestehen können. In der Regel bestimmt ein Alphabet die verfügbare Menge an Zeichen, also den Zeichenvorrat. Die Zeichen „1“, „9“, „5“ und „,“ stehen auf dieser Ebene in keinem Zusammenhang.
Durch das Zusammenfügen von Zeichen nach bestimmten Ordnungsregeln (Syntax) entstehen Daten. Daten sind für sich genommen neutral, d.h. sie stehen in keinem Zusammenhang, Kontext oder Bezug zueinander. Der Übergang von Daten zu Information ist fließend, denn Daten an sich haben keinen Typ. In der

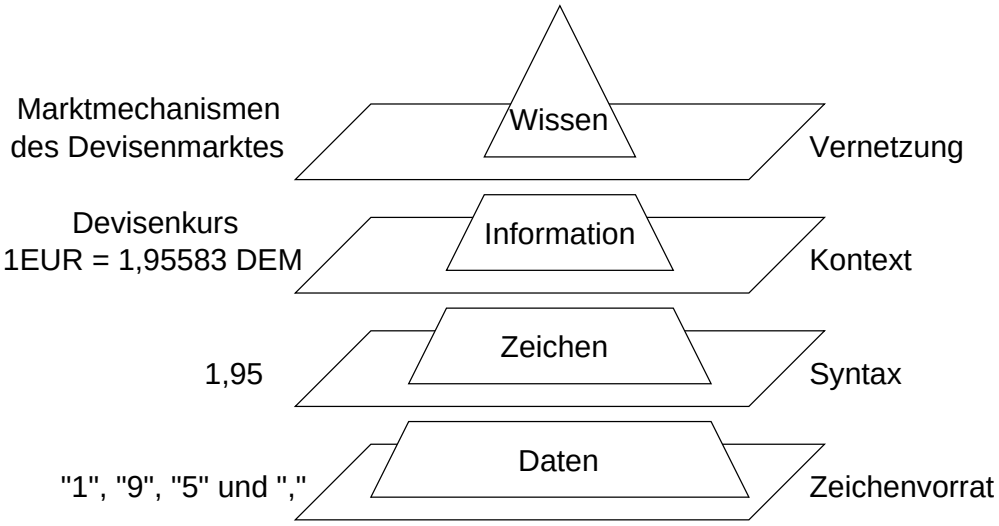


Abbildung 2.1: Begriffshierarchie oder Wissenspyramide

Regel werden Daten jedoch zusammen mit ihrem Typ in einem System gespeichert. Man könnte z.B. das Datum „1,95“ mit dem Typ Währung versehen. Diese Typangabe ist ein Metadatum und noch keine echte Information, da es keinen Empfänger gibt, der diesem Zusammenhang von Datum und Typ eine Bedeutung zukommen lassen könnte.
Eine Information ist immer das Ergebnis einer Nachricht (die aus Daten und Zeichen besteht), die von einem Empfänger verarbeitet wird und ihr eine Bedeutung gibt. D.h. Information ist ein bedeutungstragender Zusammenhang, der zwischen mindestens zwei unterschiedlichen Systemen wirkt und im Empfängersystem etwas bewirkt. Die Übertragung von Informationen durch Nachrichten ist in Abbildung 2.2 dargestellt.
In einem weiteren Schritt wird aus den Daten und Informationen Wissen gewonnen. Während Informationen zu Aktionen oder Entscheidungen führen, führt Wissen zu Handlungen und Anwendungen. Dies setzt ein Erkennen voraus. Wissen ist dabei eher ein Prozess, denn durch Hinzufügen weiterer Informationen kann sich Wissen beim Individuum ändern. Wissen hat auch jede Neutralität verloren, denn Wissen kann falsch oder richtig sein (ohne zusätzliche Informationen ist dies dem Individuum jedoch nicht bewußt). Nach [Probst et al. 1999, Seite 46] bezeichnet Wissen

> „[...] die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfaßt sowohl theoretische Kennt-

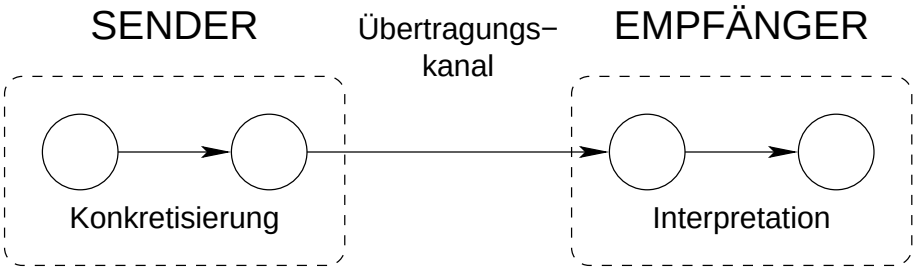


Abbildung 2.2: Modell für die Informationsweitergabe [Lagemann 1987, Seite 2]

> nisse als auch praktische Alltagsregeln und Handlungsanweisungen. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebunden. Daher müssen Daten-, Informations- und Wissensmanagement stets zusammenspielen."

Wissen lässt sich in die folgenden Kategorien einteilen (vgl. [Forst 1999] und [Nonaka et al. 1997]):

Implizites Wissen Beim impliziten Wissen handelt es sich um persönliches, kontextspezifisches und verborgenes Wissen, das aus Können, Handlungsroutinen, Überzeugen, Glaubenssätzen und geistigen Schemata besteht.

Explizites Wissen Beim expliziten Wissen liegt formales Wissen vor, das leicht durch Worte, Zahlen oder technische Angaben darstellbar ist. Explizites Wissen kann im Gegensatz zu implizitem Wissen leicht vermehrt werden, weil es sich in formaler, systematischer Sprache weitergeben läßt.

Know-how Das Know-how ist die Teilmenge des Wissens, das für die Erbringung von Leistungen relevant ist. [Forst 1999] differenziert das Unternehmens-Know-how in:

- *Humankapital*: Es beinhaltet alle individuellen Kenntnisse und Fähigkeiten der Mitarbeiter und ist personengebunden.
- *Strukturelles Kapital*: Hierbei handelt es sich um das in einem Unternehmen organisatorisch verankerte Wissen. Dieses Wissen bliebe auch dann erhalten, wenn alle Mitarbeiter auf einmal das Unternehmen verlassen würden.

Die Grenzen zwischen Daten, Informationen und Wissen sind nicht scharf, sondern fließend. Sie bilden ein Kontinuum, wobei sich ein stetiger Qualitätswandel zwischen den beiden Polen Daten und Wissen vollzieht (siehe Abbildung 2.3).

Daten		**Information**		**Wissen**
unstrukturiert	.....		.....	strukturiert
isoliert	.....		.....	verankert
kontext-unabhängig	.....		.....	kontext-abhängig
geringe Verhaltenssteuerung	.....		.....	hohe Verhaltenssteuerung
Zeichen	.....		.....	kognitive Handlungsmuster
distinction	.....		.....	mastery/capability

Abbildung 2.3: Kontinuum von Daten, Information und Wissen (vgl. [Probst et al. 1999, Seite 38])

Informationen und Wissen haben in den letzten Jahren eine starke Aufwertung erfahren. Sie werden sogar häufig schon als der 4. und 5. Produktionsfaktor in der Betriebswirtschaftslehre angesehen.[1] Sie scheinen u.a. auch deshalb so lukrativ, weil sie sich – im Vergleich zu den materiellen Produktionsfaktoren – nicht verbrauchen, sondern sogar vermehren. Allerdings sollte man nicht vergessen, dass die Bildung von Informationen und Wissen erhebliche Kosten verursachen, die kaum genau zu bestimmen sind. Trotzdem sollte in sie investiert werden, denn

> „Wissen ist der letzte verbliebene Wettbewerbsvorteil der entwickelten Industrienationen im globalen Wettbewerb". [Nohr 2000a, Seite 5]

2.2 Warum Wissensmanagement?

Die Struktur der Wissensumwelt, in der Unternehmen heute agieren müssen, ist wesentlich komplexer als noch vor einigen Jahrzehnten. Maßgeblich tragen hierzu drei eng miteinander verbundene Trends bei: explosionsartige Vermehrung, weitgehende Fragmentierung sowie zunehmende Globalisierung des Wissens.
Aus diesem Grund ist es nicht verwunderlich, dass viele Unternehmen hoffen, durch Wissensmanagement eine Reihe ihrer Probleme lösen zu können. Dabei ist die Idee des Wissensmangements an sich gar nicht neu. Auch früher wurde schon versucht, Wissen gezielt verfügbar zu machen (vergleiche hierzu [Leonhardt 1999]). Was das heutige Wissensmanagement neu macht, sind die technologischen Möglichkeiten, die heute existieren. Zu keiner Zeit konnten so viele Datenmengen auf so kostengünstige Weise gespeichert und wieder abgerufen werden. Dadurch ergibt sich aber auch die Gefahr des wahllosen Speicherns und damit das Problem, das Gesuchte aus der Masse der Informationen wieder herausfiltern zu können.

[1] In der klassischen Betriebswirtschaftslehre benötigt man die Produktionsfaktoren Arbeit, Betriebsmittel und Wertstoffe zur Produktion von Gütern.

Die Gründe für den lohnenden Einsatz von Wissensmanagement sind vielfältig. So kam es in der Vergangenheit immer wieder vor, dass – insbesondere in größeren Unternehmen mit unterschiedlichen Standorten – Projekte und Untersuchungen mit ganz ähnlichen Zielsetzungen durchgeführt wurden, ohne gegenseitiges Wissen voneinander und so auch ohne die Möglichkeit, Ergebnisse und Erfahrungen miteinander auszutauschen. Dafür ist es aber unter Umständen mit einem doppelten Kostenaufwand für das Unternehmen verbunden.
Ebenso unbefriedigend ist es, wenn Projekterfahrungen und -erkenntnisse nicht dokumentiert werden, wodurch Fehler, wenn sie als solche frühzeitig hätten wiedererkannt werden können, nicht wiederholt worden wären. In diesen Problemkreis gehört auch die Bildung von Wissensinseln. Dies geschieht, wenn verschiedene Abteilungen oder Projekte ihr Wissen (z.B. über Technologien, Kunden, Märkte) nicht austauschen. So müsste z.B. jede Abteilung ihr Wissen über Kunden separat pflegen, obwohl es viel effizienter und kostensparender wäre, wenn alle Abteilungen auf einen gemeinsamen Wissenspool zugreifen könnten. Es ist aber nicht nur von Vorteil einen Wissenspool über seine Kunden zu besitzen. Auch über die Existenz firmeneigener Patente und Erfindungen sowie über interne Wissensträger (Experten oder Spezialisten auf ihrem Gebiet) sollte sich jeder Mitarbeiter Erkenntniss verschaffen können.
An sich ist jede Wissensbasis zunächst ein großer Vorteil für jedes Unternehmen. Allerdings darf nicht nur das Sammeln von Daten im Vordergrund stehen, da sonst die Gefahr des „Information Overload“ auftritt, wo die Quantität der Daten über ihrer Qualität steht und nicht mehr optimal das gewünschte Wissen aus dem Wissenspool extrahiert werden kann. Nicht nur zu viele, sondern auch veraltete Informationen stellen ein Problem dar. Deshalb muss eine unternehmensweite Wissensbasis ständig auf dem neuesten Stand gehalten werden.
Doch nicht nur neu hinzukommendes Wissen kann Probleme bereiten, sondern auch Wissen, das unkontrolliert abfliesst, z.B. durch das Ausscheiden eines langjährig bei einer Firma angestellten Experten. Dies kann dazu führen, dass sein Wissen für dies Unternehmen für immer verloren geht, falls der Experte dieses Wissen nicht an einen weiteren Kollegen im Vorwege weitergegeben hat. Diese Kultur des „Weitergebens von Wissen“ ist insbesondere dort anzutreffen, wo hauptsächlich implizites Wissen weitergegeben werden soll.
Nicht zuletzt sollte aber auch über den „Tellerrand“ des Unternehmens hinausgeschaut werden. Dies kann sich lohnen, wenn nicht schon existierende Technologien statt sie zu lizenzieren ein zweites Mal erfunden werden sollen. In der Literatur wird dies so schön mit „das Rad neu erfinden“ umschrieben. Doch nicht nur Unternehmen, mit denen das eigene Unternehmen noch nichts zu tun hatte, sondern auch Firmen, die früher schon einmal etwas an das eigene Unternehmen verkauft

haben (sei es Dienstleistungen, Geräte o.ä.) können auch im nachhinein von Nutzen sein, vorausgesetzt man weiss von ihnen und hat vielleicht auch noch den richtigen Ansprechpartner. In diesem Fall kann einem oft kompetenter weitergeholfen werden, als von einem Unternehmen, zu dem bisher keinerlei Beziehungen existieren.

Eine einheitliche Definition von Wissensmanagement existiert nicht. Auch wird der Begriff in der Fachliteratur eher über den Begriff des Wissens erklärt. Trotzdem sollen der Vollständigkeit halber hier einige Definitionen folgen:

Definition 1 „Wissen richtig managen bedeutet primär, jedem Mitarbeiter das benötigte Wissen zur richtigen Zeit am richtigen Ort in der richtigen Qualität und im richtigen Zustand anbieten zu können."[2]

Definition 2 „*Wissensmanagement* beschäftigt sich mit den Möglichkeiten der Einflussnahme auf die Ressource „Wissen" in Organisationen. Es befasst sich mit jenem Teil der Lernprozesse, die als gestaltbar angesehen werden. Damit grenzt es sein Erklärungsinteresse von (den meisten) Konzepten des Organisationalen Lernens ab. Wissensmanagement versucht gezielt in die organisatorische Wissensbasis einzugreifen und entwickelt zu diesem Zweck Konzepte und Methoden.
Die *organisationale Wissensbasis* umfasst sämtliche Wissensbestandteile, über die eine Organisation zur Lösung ihrer vielfältigen Aufgaben verfügt. Daten, Informationen und (stark kontext- oder personengebundenes) Wissen und Fähigkeiten müssen hierbei in ihren Verknüpfungen betrachtet werden. Fähigkeiten können auf unterschiedlichen Emergenzebenen (Individuum, Gruppe, Gesamtorganisation) vorliegen und gestaltet werden."[3]

Definition 3 [Wissensmanagement beschäftigt sich damit,] „erfolgsrelevantes Wissen zu identifizieren, zu erzeugen bzw. zu entwickeln, in Verhalten umzusetzen, wobei dieses Wissen der Organisation durch geeignete Diffusions- und Integrations- bzw. Modifikationsprozesse verfügbar und somit nutzbar gemacht wird." [Pawlowsky et al. 1997]

[2] aus: http://www.uni.de/km/km2.html. Letzter Abruf am 1.8.2001.

[3] aus: http://enterprise.cck.uni-kl.de/wmk/papers/public/Wissensidentifikation. Letzter Abruf am 25.7.2001.

2.3 Bausteine des Wissensmanagements

Die *Bausteine des Wissensmanagements*, die an der Universität Genf um Professor Gilbert Probst entstanden sind (siehe [Probst et al. 1999]), sollen eine Klassifizierung von Wissensproblemen ermöglichen und so Ansatzpunkte für Interventionen in Unternehmen liefern. Der innere Kreislauf, bestehend aus den sechs Bausteinen *Wissensidentifikation*, *Wissenserwerb*, *Wissensentwicklung*, *Wissens-(ver)teilung*, *Wissensbewahrung* und *Wissensnutzung*, bildet die Kernprozesse des Wissensmanagements (siehe Abbildung 2.4). Der äußere Kreislauf, bestehend aus den Elementen Zielsetzung (Baustein *Wissensziele*), Umsetzung (die sechs Bausteine des inneren Kreislaufs) und Messung (Baustein *Wissensbewertung*) bildet einen traditionellen Managementprozess ab. Auf diese Weise wird die Wichtigkeit strategischer Aspekte sowie die Bedeutung konkreter und eindeutiger Zielsetzungen im Wissensmanagement hervorgehoben und auch der Vergleich zwischen Zielsetzung und dem dann Erreichten ermöglicht.

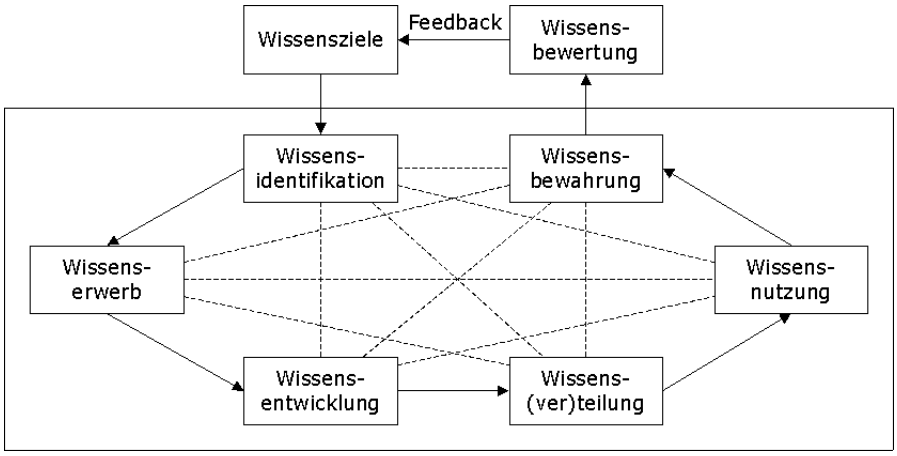


Abbildung 2.4: Bausteine des Wissensmanagements (vgl. [Probst et al. 1999, Seite 58])

Wissensziele Durch die Festlegung von Wissenszielen wird den Aktivitäten des Wissensmanagements eine Richtung gegeben. Man unterscheidet zwischen normativen, strategischen und operativen Wissenszielen:
Normative Wissensziele sind auf die Schaffung einer wissensbewussten Unternehmenskultur ausgerichtet. Dies soll durch die Teilung und Weiterentwicklung der eigenen Fähigkeiten erreicht werden.

Strategische Wissensziele definieren den zukünftigen Kompetenz- bzw. Wissensbedarf eines Unternehmens.
Operative Wissensziele konkretisieren die normativen und strategischen Wissensziele und tragen zur Umsetzung des Wissensmanagements bei. Ein typisches operatives Wissensziel könnte z.B. das Verfügbar machen aller im Unternehmen erstellten Dokumente im Intranet sein.

Wissensidentifikation Bei der Wissensidentifikation erfolgt eine Aufstellung des im Unternehmen bereits vorhandenen internen und externen Wissens und Know-hows. Dadurch soll ein Unternehmen einen Überblick über sein derzeitig bestehendes, wettbewerbsentscheidendes Wissen erhalten. Dies ist besonders in größeren Unternehmen wichtig, weil hier häufig der Gesamtüberblick über das Vorhandensein von internen und externen Daten, Informationen und Fähigkeiten nicht vorhanden ist.
Oftmals führen auch Unklarheiten darüber, wo welcher Experte mit welcher Expertise zu finden ist und an welchen Projekten derzeit gearbeitet wird bzw. an welchen früher gearbeitet wurde, zu uninformierten Entscheidungen oder Doppelspurigkeiten. Schafft man hingegen Transparenz über das Unternehmenswissen, so werden diese Fehlentwicklungen vermieden. Eine Möglichkeit, Transparenz zu schaffen, liegt in der Erstellung von Wissenskarten (siehe Kapitel 2.4). Wobei darauf geachtet werden sollte, dass nicht eine rein technologische Lösung angestrebt wird, sondern immer auch der Mitarbeiter mit einbezogen werden muss, weil er den potentiellen Wissensnutzer darstellt und damit maßgeblich über den erfolgreichen Einsatz von Wissensmanagement in einem Unternehmen entscheidet.

Die Wissensidentifikation oder -transparenz kann demnach zum Schließen von Wissenslücken genutzt werden, indem diese durch den Vergleich von bereits vorhandenem Wissen und den Wissenszielen aufgedeckt werden können.

Wissenstransparenz hat aber auch ihren Preis, denn die Erfassung oder gar Messung von Fähigkeiten kosten Zeit und Geld. Es müssen evtl. Interviews geführt werden, Fragebögen verschickt und Testverfahren entwickelt werden. Dieser Aufwand muss sich lohnen. Bessere Transparenz sollte daher nur über die erfolgskritischen Fähigkeiten einer Organisation erzeugt werden. Verfolgt man eine Strategie der „Transparenz um jeden Preis“, so macht dies keinen Sinn, sondern kann sogar kontraproduktiv sein.

Wissenserwerb Beim Wissenserwerb handelt es sich um den gezielten „Einkauf“ von Wissen, nicht aber um den Erwerb von Wissen im Sinne von Lernen.

Den Unternehmen bieten sich für diesen Erwerb verschiedene Beschaffungsstrategien an:

- *Erwerb externer Experten bzw. Wissensträger.* Fehlt es in einem Unternehmen an bestimmten Kompetenzen oder Fachwissen, so ist es oft am einfachsten, Mitarbeiter einzustellen, die über eben diese Fähigkeiten verfügen. Dies kann entweder über eine klassische Festanstellung geschehen oder – falls diese Fähigkeiten nur für einen begrenzten Zeitraum benötigt werden – über eine Zeitarbeitsfirma oder Unternehmensberatung. Auf die Schwierigkeiten bei der Rekrutierung neuer Mitarbeiter soll hier nicht eingegangen werden. Bei weiterem Interesse ist [Probst et al. 1999] zu empfehlen.
- *Erwerb von Wissen anderer Firmen.* Insebsondere Großunternehmen wählen häufig einen sehr radikalen Weg des Wissenserwebs: die Akquisition. Sie erhoffen sich, auf diese Weise ihre Wissenslücke „auf einen Schlag“ schließen zu können. Weniger radikal, kostengünstiger und risikoloser ist der Erwerb von Wissen durch *strategische Allianzen*, *Product-* und *Knowledge Links*. Hier stehen das Erlangen bestimmter Ziele, Vorteile bei Produkteinführungen und gegenseitiges Lernen und Wissenerwerb im Vordergrund.
- *Erwerb von sog. Stakeholderwissen*[4]. Kunden eines Unternehmens wissen häufig mehr über die Stärken und Schwächen eines Produktes als die Entwickler, weil sie oft tagtäglich mit ihnen arbeiten. Dieses Wissen kann sich ein Unternehmen zu nutze machen.
- *Erwerb von Wissensprodukten.* Unter dieser Rubrik versteht man den Ankauf von sog. „Wissenskonserven“, also z.B. von Software oder Patenten. Dies kann einen enormen Vorteil für ein effektives Wissensmanagement schaffen. Allerdings sollte schon im Vorfeld geklärt werden, ob sich diese Produkte auch in die Arbeitsabläufe integrieren lassen, da der Kauf allein noch keinen Vorteil bringt.

Wissensentwicklung Bei der Wissensentwicklung steht die Produktion neuer Fähigkeiten und Produkte, besserer Ideen und leistungsfähigerer Prozesse im Vordergrund. Dabei kann es sich um das Schaffen neuer Fähigkeiten in einem Unternehmen oder gar um das Kreieren auch extern noch nicht

[4]Stakeholder sind diejenigen Gruppen im Umfeld einer Organisation, die besondere Interessen und Ansprüche an die Tätigkeit eines Unternehmens richten. Konkrete Beispiele sind Kunden oder Lieferanten.

vorhandener Fähigkeiten handeln. Wissen kann auf individueller oder kollektiver Ebene entwickelt werden.
Bei der individuellen Wissensentwicklung ist es entweder Kreativität oder die Fähigkeit zur systematischen Problemlösung, die den Wissensentwicklungsprozess ausmacht. Die kollektive Wissensentwicklung ist nur dann der individuellen überlegen, wenn die Teams „gut zusammenspielen“ können, d.h. es muss eine Atmosphäre des Vertrauens und der Kommunikationsbereitschaft existieren. Dieser Prozess kann durch die Einführung von „Think tanks“[5], „Lernarenen“[6] oder „Produktkliniken“[7] unterstützt werden. Oft spielen auch beide Komponenten zusammen. Die Unternehmensführung sollte deshalb diese Fähigkeiten ihrer Mitarbeiter bestmöglich unterstützen.
Hilfreich bei der Wissensentwicklung ist auch das Erstellen von „Lessons learned“[8] nach Abschluss eines Projektes. Es muss dabei klar herausgestellt werden, welche Erfahrungen im Projektverlauf gesammelt wurden und worauf ein zukünftiges Team mit ähnlichen Aufgaben achten sollte.

Dass Wissensentwicklung lohnend sein kann, zeigt das Beispiel der Firma `3M`, die mit Produkten, die nicht älter als vier Jahre sind, über 30 Prozent ihres Umsatzes erzielt. Dies ist durchaus gewollt, denn immerhin dürfen die Mitarbeiter der Forschungsabteilung knapp einen Arbeitstag pro Woche (15% der Arbeitszeit) für die Entwicklung eigener Ideen verwenden und darüber hinaus auch auf die Unternehmensressourcen zurückgreifen. Auf diese Weise sind z.B. die Post-it-Notes entstanden. Ein ungeschriebenes Gesetz lautet sogar: „Du sollst keine Ideen für neue Produkte töten“. Psychologisch gesehen ist es für die Mitarbeiter auch ganz wichtig, dass eine Toleranz gegenüber Fehlschlägen besteht. Im Falle des Scheiterns einer Idee sind den Mitarbeitern auch weiterhin ihr Status und ihr früheres Gehalt garantiert.

[5] „In Think tanks konzentriert die Organisation ihre Intelligenz und betraut sie mit der Entwicklung kritischer Fähigkeiten für die Gesamtorganisation. Traditionelle Formen von Think tanks sind Stäbe und Forschungs- und Entwicklungsabteilungen.“ Aus: [Probst et al. 1999, Seite 206]

[6] „Als Lernarena kann eine Struktur bezeichnet werden, welche die gewöhnliche Aufbau- und Ablauforganisation eines Unternehmens überlagert, ohne diese zu ersetzen. Sie dient der Steuerung von Lernprozessen im Unternehmen und ermöglicht durch eine bewußte Auswahl von Lernträgern und Lernarenenverantwortlichen eine gezielte Wissens(ver)teilung.“ Aus: [Probst et al. 1999, Seite 241]

[7] Die Grundidee der Produktklinik liegt darin, „dass eigene aktuelle Produkte und Prozesse aufbauend auf Markt-, Wettbewerbs- und Kundendaten direkt auf physischer Ebene mit den Mitbewerbern verglichen werden.“ Aus: [Wildemann 1996, Seite 39]

[8] „Lessons learned repräsentieren die Essenz der Erfahrungen, welche in einem Projekt oder einer Position gemacht wurden.“ Aus: [Probst et al. 1999, Seite 212]

Näheres zu diesem Beispiel findet sich in [Nonaka et al. 1997, Seite 153ff] und [Probst et al. 1999, Seite 68ff].

Wissens(ver)teilung Die Wissens(ver)teilung schafft die Voraussetzung für die Wissensnutzung. Dabei kann es sich entweder um die zentral gesteuerte Verteilung organisationalen Wissens auf eine festgelegte Gruppe von Mitarbeitern oder um das direkte Mitteilen von Wissen unter Personen handeln. Nur wenn eine Unternehmung es schafft, ihre Mitarbeiter dazu zu bewegen, ihr Wissen nicht zu horten (im Sinne von „Wissen ist Macht"), sondern in geeigneter Form rechtzeitig weiterzugeben, kann effektives Wissensmanagement stattfinden. In Zeiten von Stellenabbau ist dies natürlich sehr schwierig, weil bei den Mitarbeitern die Angst herrscht, sie könnten ersetzbar werden, wenn sie ihr Wissen preisgeben. Oft sind sich Menschen aber auch gar nicht ihres Wissens bewusst. Dies geschieht insbesondere dann, wenn es sich um implizites Wissen handelt. Dieses Wissen, das in den Köpfen der Mitarbeiter existiert, ist häufig nur schwer explizierbar und somit auch nur sehr begrenzt weitergebbar. In einigen Firmen versucht man, durch Anreizsysteme die Mitarbeiter dazu zu bewegen, ihr Wissen preis zu geben. Dies ist aber insofern kritisch, wenn dieser Bonus einmal wegfallen sollte. Ein wahrscheinlich besserer Ansatz ist es, die Ideen des Wissensmanagements in die Unternehmenskultur mit aufzunehmen, was aber auch bedeutet, dass dem Gut Wissen von allen eine hohe Bedeutung beigemessen werden muss und den Mitarbeitern z.B. Zeit für die Wissensweitergabe eingeräumt werden muss.

Die zentrale Frage, um die es bei der Wissens(ver)teilung geht, lautet: Wer sollte was in welchem Umfang wissen oder können? Nicht jeder Mitarbeiter möchte oder sollte auf die gesamte Wissensbasis zugreifen können. Dies würde zum einen zu einer Überlastung des Mitarbeiters mit irrelevanten Informationen führen und somit zu einer Erschwerung des Auffindens von relevanten Informationen. Zum anderen muss die Geheimhaltung bestimmter Informationen bewahrt werden. Dies können z.B. Firmengeheimnisse oder datenschutzrechtlich geschützte Daten sein.

Ein Instrument, das den Prozess der Wissens(ver)teilung unterstützen kann, ist die Schaffung und Betreuung von persönlichen Netzwerken. Dies kann z.B. als informelles Treffen gleichaltriger Firmenmitglieder, als regelmäßige Treffen von Spezialisten oder als interdisziplinäre Netzwerke durchgeführt werden. Das entscheidende hierbei ist, eine direkte Verbindung zwischen den Mitarbeitern herzustellen, um so den Wissensaustausch zu fördern.

Wissensnutzung Bei der Wissensnutzung geht es um den konkreten Einsatz des Unternehmenswissens zum Wohle und zum Nutzen des Unternehmens. Ohne diese Wissensnutzung ist das ganze Wissensmanagement vergebens. Der potentielle Nutzer muß also vom Nutzen der neuen Lösungen überzeugt sein, um sie zu verwenden. Es gibt verschiedene Kriterien, die dabei helfen sollen, dass dieses Wissensangebot angenommen wird. Zum einen sollten die Prozesse der Wissensidentifikation, der Wissensentwicklung, des Wissenserwerbs und der Wissens(ver)teilung immer die Bedürfnisse potentieller Nutzer berücksichtigen. Zum anderen kann der Nutzer auch direkt in diese Prozesse involviert werden, um seinen Wünschen gerecht werden zu können. Weiterhin sollte das zu nutzende System der Bequemlichkeit des Nutzers entgegenkommen. Es hat sich nämlich gezeigt, dass die einfache Benutzbarkeit einen entscheidenden Einfluss auf die Nutzung hat. Aber auch die Aktualität und Richtigkeit der Daten muss gewährleistet sein, sonst kann es zu einer „Todesspirale“ kommen, in der die Nutzung des Systems weiter zurück geht und wenn keine Investionen in die Zugriffsfreundlichkeit vorgenommen werden die Datenqualität sich verschlechtert, was wiederum das Vertrauen in die Daten schwächt und seinerseits zu einer noch geringeren Nutzung des Systems führt, womit man wieder am Ausgangspunkt der Spirale angekommen wäre, die dann so weiter läuft, bis keiner das System mehr benutzt.

Wissensbewahrung Das Ziel dieses Bausteins besteht darin, einmal im Unternehmen vorhandenes Wissen nicht wieder zu verlieren. Um dies zu erreichen, müssen die Prozesse der *Selektion* des Bewahrungswürdigen, die angemessene *Speicherung* und die regelmäßige *Aktualisierung* sorgfältig durchgeführt werden.

Beim Selektionsprozess sollte man von der Leitregel ausgehen, dass nur das, was für zukünftige Nutzer interessant sein könnte, es wert ist bewahrt zu werden.

Der Speicherungsprozess findet auf drei Ebenen statt: der individuellen, der kollektiven und der elektronischen. Auf der *individuellen* Ebene versucht man die Mitarbeiter, die für das Unternehmen von großer Bedeutung sind, durch soziale oder materielle Anreizsysteme an das Unternehmen zu binden. Auf der *kollektiven* Ebene wird „Gruppenwissen“ gespeichert, das die Mitarbeiter kollektiv in Arbeitsgruppen oder einfach durch unternehmensinterne Begrifflichkeiten erlangen. Die Speicherung auf *elektronischer* Ebene sollte an sich problemlos erfolgen, wenn einige Regeln beachtet werden: Es sollte einen allgemein guten Schutz gegen Datenverlust geben, z.B.

durch Back-ups. Außerdem sollten Dateien so abgespeichert werden, dass sie auch wiedergefunden werden können. Metadaten können hierbei eine große Hilfe sein (siehe hierzu Kapitel 5).
Beim Aktualisierungsprozess muss das Unternehmen sich vor Augen halten, dass es auch hier wieder zu der weiter oben beschriebenen „Todesspirale" kommen kann, wenn die Daten in der Wissensbasis nicht aktuell sind. Eine ausführliche Aufstellung der Formen des organisationalen Vergessens findet sich in [Probst et al. 1999, Seite 317].

Wissensbewertung In diesem letzten Schritt sollen die Maßnahmen des Wissensmanagements beurteilt werden, also ob beispielsweise die Wissensziele erreicht wurden. Die größte Hürde stellt hierbei das Meßverfahren dar, da es keine allgemeingültigen Bewertungsprinzipien für das Messen von Wissen gibt. Eine Möglichkeit, mit der sowohl materielle als auch immaterielle Güter dargestellt und bewertet werden können, bietet die sog. *Balanced Scorecard* von Kaplan und Norton (siehe dazu [Kaplan et al. 1992]) oder der *Intangible Asset Monitor* [Sveiby 1997]. Beide Verfahren ergänzen die finanziellen Erfolgsgrößen eines Unternehmens um nicht-finanzielle Größen. Teilweise bieten sich auch technische Hilfsmittel zur Wissensbewertung an. Verfügt ein Unternehmen beispielsweise über ein Intranet, so kann gemessen werden, in welchem Maße auf dieses Wissen zugegriffen wird und kann so ggf. gezielt hierauf einwirken.
Ebenso ist es von Vorteil, wenn die Wissensziele so formuliert wurden, dass eine Messung des Erfolgs zumindest ansatzweise möglich ist. So sind eher vage Formulierungen wie: „Unser Unternehmen soll ein lernendes Unternehmen werden" zwar von der Zielrichtung her für die Umsetzung von Wissensmanagementmaßnahmen wichtig, aber kaum messbar.

Die Definition von Bausteinen im Wissensmanagement bietet folgende Vorteile (nach [Probst et al. 1997]):

- Strukturierung des Managementprozesses in logische Phasen,
- Ansätze für Interventionen werden geboten,
- ein Raster für die Suche nach den Ursachen von Wissensproblemen wird geliefert.

Da das Hauptaugenmerk meiner Arbeit in der Visualisierung von bereits bestehenden Wissensbeständen liegt, werde ich mich im folgenden etwas ausfühlicher dem Thema Wissenskarten widmen.

2.4 Transparenz durch Wissenskarten

Wissensmanagement und mit ihnen Wissenskarten sind besonders gut dafür geeignet, Mitarbeiter, die nach einem bestimmten Problemlösungsansatz suchen, an Mitarbeiter zu verweisen, die bereits früher schon bei analogen Problemen gute Lösungen gefunden haben. Dies trägt auch dem Umstand Rechnung, dass Wissen immer häufiger als Ressource anerkannt wird, die nicht nur von Menschen generiert, sondern auch an sie gebunden ist. Know-how beruht auf Erfahrungen und läßt sich nur sehr schwer externalisieren. Nonaka und Takeuchi widmen sich diesem Thema ausführlich in [Nonaka et al. 1997]. Auch auf Managementebene hat sich mittlerweile die Erkenntniss durchgesetzt, dass wesentliche Werte der Unternehmen „in den Köpfen der Mitarbeiter“ zu finden sind.

Nach [Eppler 1997] sind

> „Wissenskarten allgemein formuliert graphische Verzeichnisse von Wissensträgern, Wissensbeständen, Wissensquellen, Wissensstrukturen oder Wissensanwendungen. Neben der Transparenzerhöhung ermöglichen sie das Auffinden von Wissensträgern oder -quellen, erleichtern sie das Einordnen von neuem Wissen in bestehendes und verbinden Aufgaben mit Wissensbeständen bzw. -trägern.“

Bevor eine Wissenskarte generiert werden kann, müssen die benötigten Informationen in den Computer gebracht, die Daten nach unterschiedlichen Kriterien stukturiert und die Visualisierungsmöglichkeiten eruiert werden. Auf diese Weise kann der Zugriff auf formalisierbares Wissen enorm vereinfacht und für einen großen Personenkreis zeit- und ortsunabhängig zugänglich gemacht werden.

Wissenskarten können je nach ihrer Struktur in unterschiedliche Typen untergliedert werden (siehe [Eppler 1997], [Probst et al. 1999] und [Nohr 2000b]):

- Wissensträgerkarten
- Wissensbestandskarten
- Wissensstrukturkarten
- Wissensanwendungskarten
- Wissensbeschaffungskarten/Wissenserwerbskarten
- Wissensmatrix

- Geografische Informationssysteme

Im Folgenden wird auf die einzelnen Arten von Wissenskarten näher eingegangen:

Wissensträgerkarten sind Wissenskarten, die sich auf Wissensträger (Experten) im Unternehmen oder seinem Umfeld beziehen. Man kann sie mit „Gelben Seiten“ vergleichen, weil sie über Experten mit spezifischem Wissen Auskunft geben. Auf diese Weise tragen die Wissensträgerkarten dem Umstand Rechnung, dass Wissen zumeist nur sehr aufwendig explizierbar und somit auch nur schwer übertragbar ist.

In den Bereich der Wissensträgerkarten gehören auch die Wissenstopografien und Wissensquellenkarten (siehe Abbildung 2.6).

> „*Wissenstopographien* veranschaulichen, welche Wissensart [...] in welcher Ausprägung bei welchem Wissensträger vorhanden ist. So soll sich ein Wissensnachfrager relativ rasch einen Überblick verschaffen, was von wem in welchem Detaillierungsgrad gewusst oder beherrscht wird.“ [Romhardt 1998, Seite 124]

Abbildung 2.5 zeigt dies an einem fiktiven Beispiel. Hier wird die Erfahrung von Personen anhand eines Balkens dargestellt. Ist der zu einer Person und einer Programmiersprache gehörige Balken kurz, ist die Erfahrung gering, ist er lang, ist die Erfahrung eher hoch. Der Betrachter einer solchen Grafik soll in die Lage versetzt werden, sich schnell einen Überblick verschaffen zu können.

> „*Wissensquellenkarten* zeigen, welche Personen innerhalb eines Teams, innerhalb der Organisation oder im externen Umfeld, relevantes Wissen zur Bewältigung einer spezifischen Aufgabe beitragen können. Die Experten des entsprechenden Wissensfeldes werden optisch herausgehoben.“ [Romhardt 1998, Seite 124]

Wissensbestandskarten zeigen an, wo und wie bestimmte Wissensbestände gespeichert sind. Für den Nutzer macht es einen großen Unterschied, ob die gesuchten Informationen in einem Rechenzentrum, auf einer CD-ROM, in Papierform oder im Gedächtnis eines bereits pensionierten Experten zu finden sind. Damit berücksichtigen Wissensbestandskarten den Aggregationszustand des Wissens und geben dem Nutzer wertvolle Infomationen über mögliche Weiterverarbeitungsschritte.

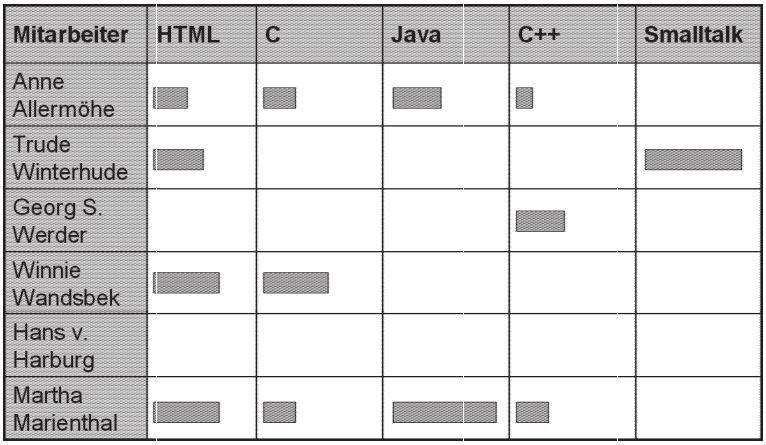


Abbildung 2.5: Wissenstopografie (in ähnlicher Form in [Probst et al. 1999, Seite 111] zu finden)

Wissensstrukturkarten machen Beziehungen, Zusammenhänge und Abhängigkeiten zwischen Sachverhalten sichtbar und liefern somit Informationen, die dem Aufgabenverständnis und der Aufgabenbewältigung dienen sollen. Anwendungsmöglichkeiten bieten sich z.B. bei der Prozessanalyse oder im Projektmanagement. Eine oft verwendete Visualisierungsmöglichkeit bieten Programme zur Erstellung von MindMaps. Eine ausführliche Beschreibung einer solchen Software findet sich in Kapitel 4.3.

Wissensanwendungskarten eignen sich besonders gut zur Visualisierung der Reihenfolge von Projekt- oder Prozessschritten und setzen diese in Beziehung zum jeweils benötigten Wissen für deren Ausführung. Sie geben außerdem noch Auskunft über die Experten oder Wissensressourcen, die für die Durchführung des jeweiligen Schrittes notwendig sind. Ein Beispiel für eine Wissensanwendungskarte zeigt Abbildung 2.7.

Wissensbeschaffungskarten oder auch Wissenserwerbskarten sollen fehlendes Wissen in einem Unternehmen aufzeigen. Auf diese Weise soll es ermöglicht werden, Wissenslücken schneller schließen zu können und eine schnellere Erfolgskontrolle durchführen zu können. Eine solche Karte kann natürlich nur erstellt werden, wenn es vorher eine Bestandsaufnahme des vorhandenen Wissens in einem Unternehmen gegeben hat.

Wissensmatrix In einer Wissensmatrix können – je nach Fragestellung – beliebige Wissensbestände oder Fähigkeiten im Verhältnis zu zwei Spannungs-

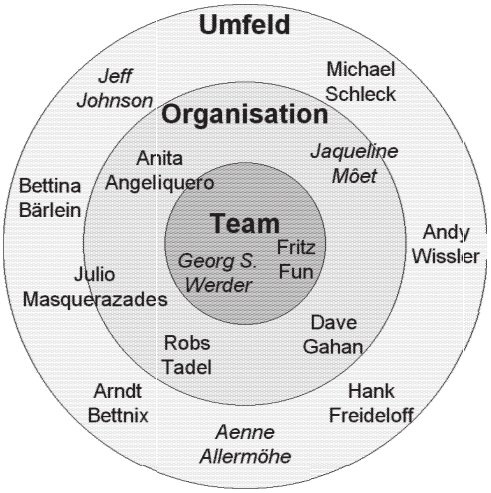


Abbildung 2.6: Wissensquellenkarte (in ähnlicher Form in [Probst et al. 1999, Seite 112] zu finden)

feldern positioniert werden. Abbildung 2.8 zeigt dies an einem Beispiel, in dem jedem Mitarbeiter zu bestimmten Fähigkeiten eine geringe (G), mittlere (M) oder hohe (H) Qualifikation zugeordnet wird. So erhält der Betrachter einen schnellen Überblick über die allgemeine Qualifikation und kann gleichzeitig den Qualifikationsgrad einer bestimmten Person erkennen.

Geografische Informationssysteme (GIS) sind den klassischen Landkarten nachempfunden und dienen der Darstellung von geografisch angeordneten Wissensbeständen. Sie ordnen Informationen nach geografischen Kriterien zu. So können beispielsweise zur Unterstützung von Marketingmaßnahmen Informationen über Verkaufsregionen geografisch dargestellt werden. Diese intuitive Darstellungsweise kann die Effektivität von Managemententscheidungen enorm erhöhen.

Wissenskarten können eine Reihe von positiven Einflüssen im Rahmen des Wissensmanagements ausüben (vgl. [Nohr 2000b, Seite 7]):

- Sie verhelfen zu einer transparenten Organisation des vorhandenen Wissens innerhalb einer Unternehmung („Was wissen wir?“).
- Sie unterstützen den Such- und Findevorgang und helfen somit die benötigten Kompetenzen im Unternehmen zugänglich zu machen und damit Zeit bzw. Aufwand zu reduzieren („gewusst wo“).

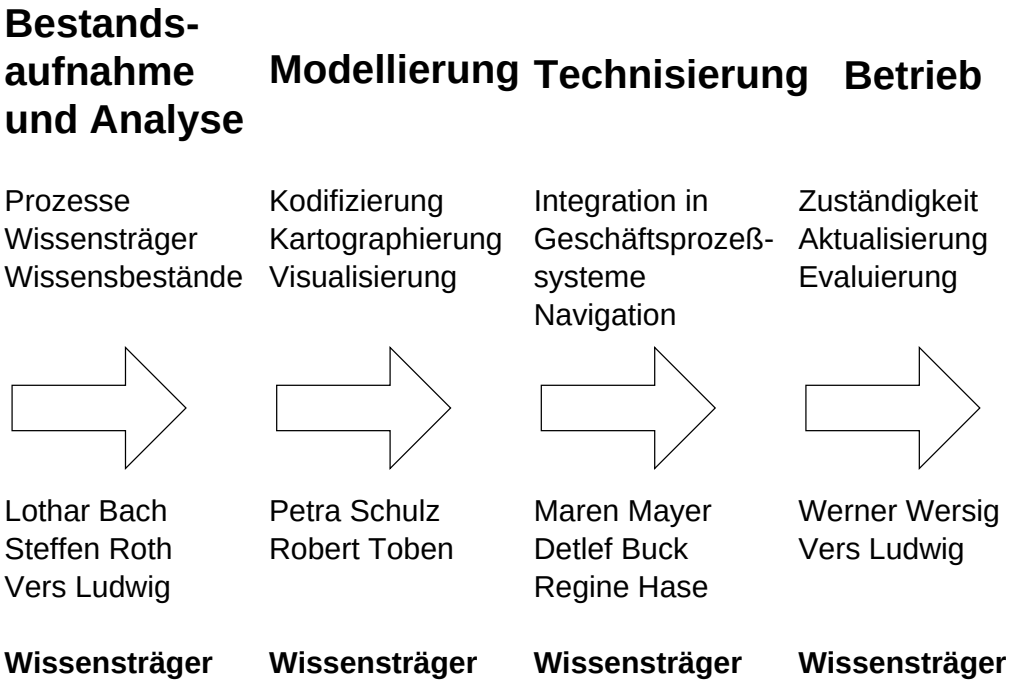


Abbildung 2.7: Wissensanwendungskarte (aus: [Nohr 2000b, Seite 16])

- Sie zeigen Lücken und Schwächen im Wissensbestand einer Unternehmung auf und sind damit Auslöser der Generierung oder Beschaffung von Wissen bzw. von Prozessen und Maßnahmen des Organisationalen Lernens („Was müssen wir lernen?“).
- Sie dienen der Erfolgsmessung im Fortschritt von Wissensmanagement- und Lernprojekten („Was haben wir gelernt?“).
- Sie können Aufgaben im Rahmen von Geschäftsprozessen und Wissensträger miteinander verzahnen.
- Sie helfen, Wissensstrukturen transparent zu machen und somit Zusammenhänge und Abhängigkeiten aufzudecken.

Wissenskarten haben aber auch kritische Seiten (vgl. [Probst et al. 1999, Seite 114ff] und [Nohr 2000b, Seite 8]:

- Sie verschieben die internen Machtverhältnisse durch die Popularisierung von Wissen, das sich auf tiefere Hierarchien verschiebt.

Produkte	Router				Hubs			NMS		NOS			...
Mitarbeiter	Router A	Router B	ISDN Router	Remote Access	Hub A	Hub B	Hub C	NMS A	NMS B	NOS A	NOS B	NOS C	...
A. Abel	G	H			M		M			G			...
B. Blum	T	G				G	M				M		...
C. Claus				G		M		T	M		H		...
D. Durst		H			H					M			...
E. Eder	M		G				H		M			H	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Abbildung 2.8: Beispiel einer Wissensmatrix (nach [Probst et al. 1999, Seite 348])

- Sie sind im Ansatz eine statische Abbildung der Wissenslandschaft eines Unternehmens. Sie müssen daher als lebendige Dokumente verstanden werden, die niemals fertig werden und ständig weiterentwickelt werden müssen. Die Qualität der Daten und Informationen der Wissenslandkarten ist der entscheidende Faktor für den Erfolg und die Nutzungsbereitschaft des Systems. Daher sollte man sich insbesondere in der Startphase bewusst auf Wissensbestandteile mit hohem Informationsnutzen beschränken und eine Kopplung an bereits bestehende Systeme vornehmen.

- Sie dürfen die Privatsphäre oder den Datenschutz nicht verletzen.

- Sie sind nur schwer in Phasen des Arbeitsplatzabbaus zu realisieren, da die Angst um den eigenen Arbeitsplatz die Explizierungsbereitschaft stark hemmt.

- Sie müssen auf eine Kollektivsprache zurückgreifen können, welche die unterschiedlichen Wissensfelder integriert. Nur ein kontrolliertes Vokabular gewährleistet die vergleichbare Begriffsverwendung und Klassifikation von Informationen der unterschiedlichen Wissensdimensionen.

- Sie entfalten ihre volle Wirkung erst, wenn der Ressource Wissen bereits ein gewisser Wert zugemessen wird. Am produktivsten sind Wissenslandkarten, wenn ein effizienter Wissensmarkt existiert.

- Der Aufwand für die Erstellung und Führung von Wissenskarten ist vergleichsweise hoch.
- Wissenskarten beinhalten wichtige Informationen, die auch von Konkurrenzunternehmen ausgewertet werden können (beispielsweise bei der Abwerbung von Experten).

Ein Unternehmen, das erfolgreich Wissenskarten einsetzt, ist der Schweizer Chemiekonzern `Hoffmann-LaRoche` (vgl. [Seemann 1997]). Zuerst wurden die speziellen Kenntnisse der eigenen Forscher auf der ganzen Welt in sog. Expertenverzeichnissen erhoben. Diese Informationen wurden ähnlich einem Telefonbuch zusammengefasst und als sog. „Gelbe Seiten“ in der Organisation verteilt. Der Auflistung von typischen Problemen der Produktentwicklung wurden die Namen potentieller Problemlöser in der Organisation zugeordnet. So gewannen die Forscher einen wesentlich einfacheren Zugriff auf die interne Expertise. „Wissensinseln“ wurden verbunden und die Suchkosten nach geeigneten Ansprechpartnern für spezielle Fragestellungen konnten erheblich gesenkt werden.

Kapitel 3

Evaluationsmethoden

Bevor ich in den Kapiteln 4 und 9 auf bestimmte Evaluationsmethoden zurückgreife, soll dieses Kapitel einen kurzen Einblick in dieses Themengebiet liefern. Auf diese Weise ist es für mich auch einfacher zu erklären, warum ich welche Methode zu welchem Zweck eingesetzt habe.

Evaluation bedeutet im Deutschen soviel wie Bewertung oder Beurteilung. Evaluation kann zum Vergleich von Systemen mit ähnlicher Anwendung oder zur Überprüfung eines fertigen Systems eingesetzt werden. Idealerweise sollte Evaluation aber den gesamten Systementwicklungsprozess begleiten. Dies ist umso bedeutender, als es sehr viel kostengünstiger ist, eine Anpassung in einem frühen als in einem späten Systementwicklungsstadium zu erreichen.
Nach [Dix et al. 1998, Seite 406] hat Evaluation drei Hauptziele:

- Das Ausmaß der Systemfunktionalität zu bewerten,
- die Auswirkungen des Interfaces auf den Benutzer zu bewerten und
- die spezifischen Probleme mit dem System aufzudecken.

Evaluationsmethoden kann man in die folgenden fünf Klassen unterteilen (nach [Preece 1993, Seite 108]):

- Analytische Evaluation
- Evaluation durch Experten
- Evaluation durch Beobachtung

- Evaluation durch Befragung
- Experimentelle Evaluation

3.1 Analytische Evaluation

Diese Methode kann vom Entwicklungsstadium des Systems ausgehend vor allen anderen Evaluationsmethoden verwendet werden. Man benötigt eine formale oder semi-formale Interface-Beschreibung bzw. Spezifikation des Systems. Die Tester sind Experten und somit keine typischen Benutzer. Sie können deshalb nur Vermutungen über die kognitiven Arbeitsabläufe der Benutzer anstellen.
Die Evaluatoren erhalten nur eine beschränkte Anleitung darüber, wie evaluiert werden soll. Meist müssen sie eine Reihe von Aufgaben mit Hilfe des zu evaluierenden Systems lösen. Dies führt oft zu einem begrenzten Fokus auf das System und zu einem Mangel an diagnostischem Output für ein Redesign des Systems.
Analytische Evaluationsmethoden bieten den Vorteil, dass sie keine kostspieligen Prototypen und Benutzungstests erfordern.
Der Nachteil dieses Evaluationstyps liegt darin, dass er spezielle psychologische Kenntnisse erfordert und zeitaufwendig werden kann.

3.2 Evaluation durch Experten

Der Untersuchung liegt eine Spezifikation oder ein Prototyp zugrunde. Involviert sind Experten, um das Interface zu bewerten. Abhängig von der jeweils benutzten Methode versucht der Evaluator mehr oder minder stark das Verhalten der Benutzer zu imitieren, um so Schlussfolgerungen auf Benutzungsprobleme ziehen zu können. Die Vorteile dieser Methoden liegen darin, dass sie meist einen Überblick über die gesamte Benutzungsschnittstelle liefern und zu den gefundenen Fehlern und Problemen oft schon ein Verbesserungsvorschlag geliefert wird. Darüber hinaus werden nur relativ wenige Ressourcen benötigt, was diese Methode kostengünstig macht. Doch sind Experten leider keine Benutzer und können das wahre Benutzerverhalten deshalb auch nicht wiedergeben.Aus diesem Grund sollte die Evaluation durch Experten auch nicht als einzige Methode eingesetzt werden, sondern eher als Ergänzung zu anderen Methoden genutzt werden.
Folgende Punkte sollten bei der Auswahl der Experten beachtet werden:

- Um ein unvoreingenommenes Ergebnis zu erhalten, sollten die Experten weder an der Entwicklung von Vorgängern des Systems noch an der Entwicklung des Prototypen mitgewirkt haben.

- Die Experten sollten über ein angemessenes Fachwissen verfügen.
- Die Rolle der Experten muss klar definiert sein, damit sichergestellt ist, dass die gewünschte Benutzerperspektive von ihnen eingenommen werden kann.
- Die Aufgaben, die bearbeitet werden, und die Materialien, wie Handbücher oder Tutorials, sollten die Gleichen sein, mit denen auch die späteren Benutzer arbeiten werden.

Ein Vorteil, den die Evaluation durch Experten bietet, sind die – im Vergleich zu Tests mit Benutzern – geringen Kosten. Ausserdem liefern Experten häufig schon Lösungsvorschläge für die von ihnen entdeckten Probleme.
Es gibt aber folgende Nachteile: Experten haben bestimmte Vorlieben und bringen diese natürlich auch in die Evaluation mit ein. Ebenso ist es oft schwierig, einen sog. „doppelten Experten“ zu finden, d.h. eine Person die sich mit Softwareergonomie und mit der Art des zu untersuchenden Systems auskennt. Damit der Experte sich in den Benutzer hineinversetzen kann, benötigt er ein hohes Maß an Informationen über den Kenntnisstand der Benutzer, ihre typischen Aufgaben und ihre Reaktion auf Probleme. Eine Expertenevaluation kann niemals das weite Spektrum des Benutzerverhaltens wiedergeben, da diese des öfteren unerwartete Dinge tun.

Heuristische Evaluation

Bei der *Heuristischen Evaluation*, die von Jacob Nielsen und Rolf Molich [Molich et al. 1990] entwickelt wurde, untersuchen verschiedene Evaluatoren unabhängig voneinander das System und notieren die von ihnen gefundenen Fehler und Schwächen. Durch diese Vorgehensweise werden von den einzelnen Personen unterschiedliche Fehler gefunden. In einer Fallstudie hat sich z.B. gezeigt, dass bereits fünf Evaluatoren 75% der Fehler aufdecken konnten. Eine größere Anzahl von Testern verbesserte das Ergebnis nicht in hohem Maße.
Eine Heuristische Evaluation kann von Entwicklern, Softwareergonomen ohne bereichsspezifische Kenntnisse (einfache Spezialisten), Softwareergonomen mit bereichsspezifischen Kenntnissen (doppelte Experten) und Benutzern durchgeführt werden. So heterogen die Gruppe der möglichen Tester sein kann, so unterschiedlich ist auch der Erfolg ihrer Evaluation. Deshalb sollten Benutzer auch eher nicht zur Heuristischen Evaluation herangezogen werden. Folgende Faktoren wirken sich auf den Erfolg aus: Erfahrung (ist der Tester ein Anfänger oder doch eher ein Experte), Häufigkeit der bisherigen Evaluationen (mit jeder weiteren Evaluation wird ein Evaluator besser) und die Tagesform. Ein ausführlicher

Performance-Vergleich zwischen den verschiedenen Evaluatorgruppen findet sich in [Nielsen 1994].
Nielsen und Molich geben in [Molich et al. 1990] einige Gestaltungsprinzipien vor, die ein Evaluator bei seiner Arbeit beachten sollte:

Einfache und natürliche Dialoge Dialoge sollten keine irrelevanten oder wenig benötigten Informationen enthalten. Jegliche Information sollte in einer natürlichen und logischen Ordnung erscheinen.

Sprich die Sprache des Benutzers Der Dialog sollte in klaren Worten, Phrasen und Konzepten ausgedrückt werden, die sich eher am Benutzer als am System orientieren.

Minimiere die Gedächtnislast des Benutzers Das Kurzzeitgedächtnis der Benutzer ist begrenzt. Deshalb sollten sie sich nicht an Informationen erinnern müssen, die in einem ganz anderen Bereich des Dialogs von Bedeutung waren. Außerdem sollten Informationen so lange auf dem Bildschirm sichtbar bleiben, bis sie nicht mehr benötigt werden.

Sei konsistent Arbeitsschritte, die ein Benutzer in einem Teil des Systems gelernt hat, sollten nicht an anderer Stelle, wo eine ähnliche Aufgabe zu lösen ist, plötzlich auf andere Art und Weise zu lösen sein.

Liefere Feedback Benutzer sollten vom System in möglichst kurzer Zeit ein angemessenes Feedback in Bezug auf ihre ausgeführte Aktion erhalten.

Liefere klar markierte Ausgänge Ein System sollte Benutzer nie in Situationen geraten lassen, aus denen sie nicht wieder zurück- bzw. herausfinden.

Liefere Shortcuts Features, die es einem einfach machen, ein System zu erlernen, sind für den erfahrenen Benutzer häufig lästig. Gut gewählte Abkürzungen - die der Anfänger nicht sieht - können in ein System aufgenommen werden, so dass beide Benutzergruppen zufrieden sind.

Liefere gute Fehlermeldungen Gute Fehlermeldungen sind defensiv, präzise und konstruktiv. Defensive Fehlermeldungen schieben das Problem auf Systemschwächen und kritisieren niemals den Benutzer. Präzise Fehlermeldungen geben dem Benutzer genaue Informationen über die Ursache des Problems. Konstruktive Fehlermeldungen geben dem Benutzer nützliche Hinweise darauf, was er als nächstes tun kann.

Vermeide Fehler Besser als die beste Fehlermeldung ist ein Design, das das Eintreten von Fehlern erst gar nicht zulässt.

Der Nachteil dieser Methode liegt darin, dass ein schwerer Fehler den gleichen Stellenwert wie ein eher unbedeutender erhält, da nur die Fehleranzahl betrachtet wird. Außerdem können leicht programmspezifische Probleme übersehen werden, die durch die Gestaltungsprinzipien nicht abgedeckt werden.

Guideline-Überprüfung

Diese Methode verwendet Benutzbarkeitsrichtlinien, um zu überprüfen, ob ein System diese erfüllt. Hierbei kann es sich z.B. um Styleguides wie den `Windows`-[Microsoft 1995] oder den `Apple`-Styleguide [Apple 1992] handeln.Bei den Vorgaben, die diese Richtlinien machen, handelt es sich zumeist um Fragen des Bildschirmaufbaus, die Tastenbelegung oder den Menüaufbau. Anhand dieses Testverfahrens lässt sich relativ schnell sagen, ob ein System grundlegende Designmängel aufweist und den Standards der getesteten Umgebung entspricht. Weil die Untersuchung aber nur unter künstlichen Bedingungen stattfindet, werden aufgabenspezifische Schwächen, die bei der realen Benutzung des Systems auftreten, leicht übersehen.
Da diese Richtlinien oft sehr umfangreich und komplex sind, erfordert die Guideline-Überprüfung vom Evaluator ein hohes Sachverständnis und wird aus diesem Grund auch nur selten in der Praxis angewendet [Mack et al. 1994, Seite 5].

Cognitive Walkthrough

Beim *Cognitive Walkthrough* [Wharton et al. 1994] versetzen sich Experten in die Rolle von Benutzern und versuchen, aus dieser Perspektive heraus, typische Benutzeraufgaben mit dem zu evaluierenden System zu erledigen. Ein Hauptanliegen des Cognitive Walkthrough ist es, herauszufinden, wie einfach ein System durch „Ausprobieren“ zu Erlernen ist. D.h. ohne Zuhilfenahme von Handbüchern oder eine explizite Einführung oder Schulung durch Dritte.
Um einen *Cognitive Walkthrough* durchführen zu können, braucht man:

- Keinen vollständig implementierten Prototypen, aber zumindest eine detaillierte Beschreibung.
- Eine Beschreibung der Aufgaben, die zukünftige Benutzer mit diesem System erledigen sollen und einen konkreten Ablauf ihrer Arbeitsschritte.
- Eine Benutzerbeschreibung, die dem Evaluator helfen soll, einschätzen zu können, welches Vorwissen und welche Erfahrungen die Benutzer mitbringen.

Bei der Untersuchung sollte sich der Evaluator ständig folgende Fragen vor Augen halten:

- Ist es dem Benutzer möglich, im Voraus zu erkennen, welchen Effekt seine Aktion auslösen wird?
- Wird der Benutzer in der Lage sein zu erkennen, dass eine von ihm benötigte Handlung zur Verfügung steht?
- Wenn der Benutzer die korrekte Funktion im Interface gefunden hat: weiß er dann auch, dass es sich um die richtige handelt?
- Versteht der Benutzer das Feedback, das er vom System erhält?

3.3 Evaluation durch Beobachtung

Bei dieser Methode steht das Beobachten oder Überwachen des Benutzers, während er ein System testet, im Vordergrund. Diese Methode bietet den Vorteil, dass Schwierigkeiten mit dem System schnell aufgedeckt werden und auch allgemein qualitativ hochwertige Daten geliefert werden. Mindestanforderung an das System ist eine Simulation, ein Prototyp oder ein Mock-Up, das vom Benutzer getestet wird. Negativ an dieser Methode ist, dass die Analyse der Daten sehr zeit- und ressourcenaufwendig sein kann.

Die Evaluation durch Beobachtung kann man in folgende Methoden unterteilen:

Direkte Beobachtung

Hierbei wird der Benutzer offenkundig von einem Evaluator während des Tests beobachtet und der Evaluator macht sich Notizen. Diese künstliche Atmosphäre kann das Benutzerverhalten beeinflussen und damit auch das Testergebnis.

Videoaufzeichnung

Das Benutzerverhalten wird mittels einer oder mehrerer Kameras aufgezeichnet und die Aufzeichnungen werden hinterher durch einen Experten ausgewertet.

Elektronische Datenaufzeichnung (Logfile-Recording)

Das *Logfile-Recording* wird verwendet, um die Interaktion zwischen Benutzer und System elektronisch aufzuzeichnen. Dazu werden sämtliche Benutzeraktionen, wie

Tastatureingaben, Mausbewegungen usw., protokolliert und in einer Datei gespeichert. Die Daten, die man durch die Aufzeichnung erhält, sind auf den ersten Blick objektiv. Da die Daten an sich aber gar nichts aussagen, muss eine Person diese Daten – unter bestimmten Gesichtspunkten und Fragestellungen – auswerten und dabei geht zwangsläufig ein gewisses Maß an Objektivität verloren.
Die Auswertung ist zumeist sehr zeitaufwendig und mühsam, weshalb häufig Hilfsmittel verwendet werden, z.B. indem die Mausbewegungen, die der Benutzer gemacht hat, alle als Striche in dem Anwendungsfenster darstellt werden. Dadurch kann man relativ schnell sehen, in welchen Bereichen der Benutzer sehr aktiv war. Eine andere Möglichkeit besteht darin, sich die Aufzeichnung der Daten noch einmal abspielen zu lassen und so zu sehen, welche Mausbewegungen oder -clicks der Anwender getätigt hat. Vorteilhaft an dieser Methode ist die Unaufdringlichkeit dem Benutzer gegenüber, der das System völlig ungestört testen kann.

Interaktive Überwachung

Ist eine Variante der direkten Beobachtung und wird auch „Wizard of Oz“ genannt. Die Besonderheit liegt darin, dass es einen „unsichtbaren“ Operator gibt, der entweder den gesamten Systemoutput simuliert oder helfend in das Geschehen eingreift, um Systemmängel zu überbrücken. Zum ersten Mal – der Begriff „Wizard of Oz“ war noch gar nicht geboren – wurde dieses Verfahren von John Gould, John Conti und Todd Hovanyecz [Gould et al. 1983] eingesetzt, um eine Textverarbeitung mit Spracheingabe zu testen. Das System erlaubte die kontinuierliche Spracheingabe von bis zu 80 Worten pro Minute. Unglücklicherweise konnten selbst die besten Sprachverarbeitungssysteme dies nicht leisten. Aus diesem Grund setzte man an einen zweiten Computer – für die Testperson unsichtbar – eine Sekretärin, die die gesprochenen Worte eintippte, die dann sowohl auf dem Bildschirm der Testperson als auch auf ihrem Kontrollbildschirm erschienen.
Wichtig ist, dass der Benutzer während des Tests nicht merkt, dass er die Antworten nicht vom Computer, sondern von einem Menschen erhält. Die Methode kann gut eingesetzt werden, um kostengünstig verschiedene Varianten eines Systems zu erproben.

Verbale Protokolle (Audioaufzeichnung)

Es wird eine Aufzeichnung der vom Benutzer während des Tests geäußerten Bemerkungen gemacht. Problem: Benutzer haben oft Probleme, ihre Schwierigkeiten in Worte zu fassen.

Lautes Denken

Bei der Methode des *Lauten Denkens* lösen repräsentative Benutzer typische Aufgaben mit dem zu evaluierenden System und äußern dabei laut ihre Gedanken. Die Evaluationsleitung befindet sich mit im Raum, hält sich aber im Hintergrund. Ihre Aufgabe besteht zum einen in der Protokollierung des Tests, die entweder eine schriftliche Notiz, eine Tonband- oder Videoaufzeichnung, ein Logfile-Recording oder ähnliches sein kann. Zum anderen muss der Benutzer von ihm im Zweifelsfall dazu animiert werden, mit dem lauten Denken fortzufahren. Im Notfall können auch Fragen vom Leiter gestellt werden, die dann aber sehr allgemein gestellt werden sollten, um das Testergebnis so wenig wie möglich zu verfälschen.
Da es vielen Mensch schwer fällt, über einen längeren Zeitraum „Selbstgespräche“ zu führen, bietet ein Test mit zwei Teilnehmern pro Sitzung hier Abhilfe. Beide Personen führen die Evaluation gemeinsam durch und erzählen sich gegenseitig, was passiert, was sie empfinden und was sie als nächstes warum tun werden.

3.4 Evaluation durch Befragung

Diese Methode versucht, die subjektive Meinung eines Benutzers über ein Interface zu eruieren. Auch hier benötigt man wieder – wie bei allen Evaluationsmethoden, die mit Benutzern durchgeführt werden – als Mindestanforderung an das System eine Simulation, besser noch einen Prototypen. Ein allgemeiner Nachteil dieser Methode ist das Problem vieler Benutzer, ihre Erfahrungen mit dem System in Worte zu fassen, was zu Fehlinterpretationen beim Testleiter führen kann. Außerdem bewerten die Tester das System im Nachhinein manchmal anders, wichtige Details werden vergessen und zuvor kaum lösbare Aufgaben werden als eher unbedeutend bewertet. Deshalb sollte so wenig Zeit wie möglich zwischen dem Systemtest - vorausgesetzt, das System ist nicht sowieso schon bekannt – und der Befragung liegen. Die Methode kann – insbesondere bei Fragebögen – ohne großen Mehraufwand mit vielen Benutzern durchgeführt werden, was zu einem sehr repräsentativem Ergebnis führt. Allerdings sollte man immer beachten, dass sowohl der Interviewer gern seine Vorlieben in das Interview hineinbringt und auch die Interviewten Beantwortungs-Vorlieben haben. Dies zeigt sich teilweise bei Fragebögen, wo eher unentschlossene Menschen gerne den „Mittelwert“ ankreuzen. Interviews sind in der Regel relativ schnell auszuwerten, was dazu führt, dass mehr Tests durchgeführt werden können als bei anderen Testmethoden. Gibt es bei der Befragung allerdings keine fest vorgegebenen Beantwortungsmöglichkeiten, so kann die Analyse der Antworten kompliziert und langatmig werden.

Strukturiertes Interview

Die Fragen sind schon im Voraus genau festgelegt und werden dann nur noch nacheinander durchgegangen. Die Person, die den Test durchführt, braucht normalerweise keine hohe Qualifikation. In öffentlichen Umfragen findet sich diese Methode sehr häufig.

Flexibles Interview

Themengebiete sind hier vorgegeben, aber keine festen Fragestellungen. Das Interview entwickelt sich auf diese Weise abhängig von den Antworten des Befragten und des Befragers. Damit ist es dem Interviewer möglich, auf interessante Details, die sich während des Interviews ergeben, näher einzugehen. Schwächen des Fragebogens können auf diese Weise zusätzlich abgeschwächt werden.

Fragebögen mit offenen Fragen

Die Person, die den Fragebogen ausfüllt, kann ihre Antworten frei formulieren. Dies macht eine Auswertung sehr schwierig und langwierig, gibt dem Befragten aber auch die Möglichkeit, alles, was er über den Test berichten möchte, zu notieren.

Fragebögen mit geschlossenen Fragen

Verschiedene Antwortmöglichkeiten sind vorgegeben, die dann angekreuzt werden müssen. Keine der vorgestellten Evaluationsmethoden liefert auf so schnelle und preiswerte Art eine derart große Menge an Ergebnissen wie diese.

3.5 Experimentelle Evaluation

Bei dieser Methode gibt es zu Beginn eine Hypothese oder Behauptung, die mit wissenschaftlich korrekten Mitteln bewiesen werden soll. Dazu werden bei den Tests bestimmte Parameter verändert, um zu beobachten, wie sich dies auf den einzelnen Test und den Testverlauf auswirkt. Besonderes Augenmerk muss auf die Auswahl

- der Teilnehmer (was aber auch für alle anderen Methoden gilt),
- der zu testenden und zu verändernden Parameter und
- der Hypothese, die getestet werden soll,

gelegt werden.
Die Teilnehmer sollten entweder die tatsächlichen (späteren) Benutzer sein oder ihnen zumindest in relevanten Punkten, wie dem Alter, der Erfahrung mit dem Computer und den Aufgabenbereichen, die sie erledigen, ähneln.
Folgendes sollte bei dieser Methode Beachtung finden:

- Durch Stichproben sollte die Zuverlässigkeit der Ergebnisse überprüft werden.
- Die Zielsetzung des Experiments sollte in Begriffe wie
  - Was hat sich verändert?
  - Was ist konstant geblieben?
  - Was ist gemessen worden?

  gefasst werden.
- Die Hypothesen sollten so spezifiziert sein, dass sie getestet werden können.

Kapitel 4

Wissensmanagement-Systeme

Ziel dieses Kapitels ist es, einige Programme, die zur Unterstützung von Wissensmanagementprozessen eingesetzt werden können, vorzustellen. Im Rahmen dieser Arbeit konnten allerdings keine kostspieligen Programme – die leicht 50.000 DM und mehr kosten – angeschafft werden. Aus diesem Grund war ich auf kostenlose Demoversionen angewiesen, die von der überwiegenden Zahl der Firmen, die solche Software herstellen, erst gar nicht angeboten werden. Nach langem Suchen habe ich dann aber drei Anwendungen gefunden, die in das Gebiet passen: Eine „echte“ Wissensmanagement-Anwendung namens *SERPersonalBrain*, ein Programm zur Erstellung von sog. Hyperbolic Trees, das *Inxight Tree Studio* und ein Programm zur Erstellung von Mind Maps, den *Mind Manager*. Die letzten zwei Programme habe ich mit in meine Untersuchungen aufgenommen, weil die Grafiken, die man mit diesen Programmen erstellen kann, gut zur Erstellung von Wissenskarten (siehe Kapitel 2.4) verwendet werden können.

Evaluiert wurden die Systeme mit der Methode der *Heuristischen Evaluation* (siehe Seite 33). Dies erschien sinnvoll, weil die Methode zum einen aus dem Bereich der Expertenevaluation stammen musste, denn diese Tests sollten ja weder von einfachen Benutzern, noch aus Sicht der Benutzer betrachtet werden. Zum anderen konnten durch die vorgegebenen Gestaltungsprinzipien alle Systemene nach gleichen Prinzipien untersucht werden.

Der unterschiedliche Umfang der einzelnen Evaluationen lässt sich nicht etwa dadurch erklären, dass einige Untersuchungen weniger ausführlich durchgeführt worden wären als andere, sondern dass unterschiedlich viele Benutzbarkeitsprobleme aufgedeckt werden konnten. So scheint das Programm *MindManager* in Bezug auf die Benutzbarkeit mit seinen zehn Fehlern besser gelungen als das Programm

SERPersonalBrain mit 46 (bei der Anwendung *Retrieve*) bzw. 61 Fehlern (bei der Anwendung *Classify*).

4.1 SERPersonalBrain

Die Software *SERPersonalBrain* des Unternehmens `SER Systems AG` erlaubt die Suche in Datenbeständen auf der lokalen Festplatte (dies beinhaltet das Filesystem, die Mailverzeichnisse und Webseiten) sowie die Klassifizierung dieser Dokumente in vom Anwender definierten Kategorien. In beiden Fällen wird dabei die Knowledge Management Technologie SERbrainware genutzt. Für die beiden Aufgaben stehen die beiden Programme *Retrieve* und *Classify* zur Verfügung. Die durch sie zusammengestellten Wissensbasen solln es besonders einfach machen, bei einer Suche relevante Dokumente wiederzufinden. Dazu wird die SERware-Suchmaschine verwendet, die mit neuronalen Netzen und assoziativer Suche nach Dokumenten fahndet, die ähnliche Kombinationen von Schlagworten enthalten.
Die SER Systems AG erlaubt die kostenlose Nutzung ihres Programms bis Ende August 2001.[1] Grundlage dieser Evaluation ist die Version 1.02a.

4.1.1 Retrieve

Die Anwendung *Retrieve* erlaubt die Suche nach Dokumenten auf der eigenen Festplatte anhand von vom Anwender angegebenen Beispielen. Dies ist auch gleich der große Unterschied gegenüber herkömmlichen Suchprogrammen wie z.B. die in Windows integrierte Suche nach Dateien (Start → Suchen → Dateien/Ordner). Der Anwender gibt nicht einen Suchbegriff ein oder verwendet mehrere Suchbegriffe mit booleschen Kombinationsoperatoren, sondern gibt einen Beispielsatz oder -absatz ein.
Abbildung 4.1 zeigt die Benutzungsoberfläche von *SERPersonalBrain Retrieve*.

Untersuchung der Benutzungsoberfläche

Die englischsprachige Oberfläche hat ein modern-metallisches und - meines Empfindens nach – platzverschwenderisches Aussehen. Hinter den vier Buttons links stehen die Funktionen `Konfiguration` (Schraubenschlüssel), `Suche` (Lupe), `internes Anzeigeprogramm` (Fenstersymbol) und `Online-Hilfe` (Fragezeichen). Die Funktionen `Konfiguration` und `Online-Hilfe` öffnen jeweils ein neues Dialogfenster, während die Funktion `internes Anzeigeprogramm` ein Umschalt-But-

[1] http://www.ser.de

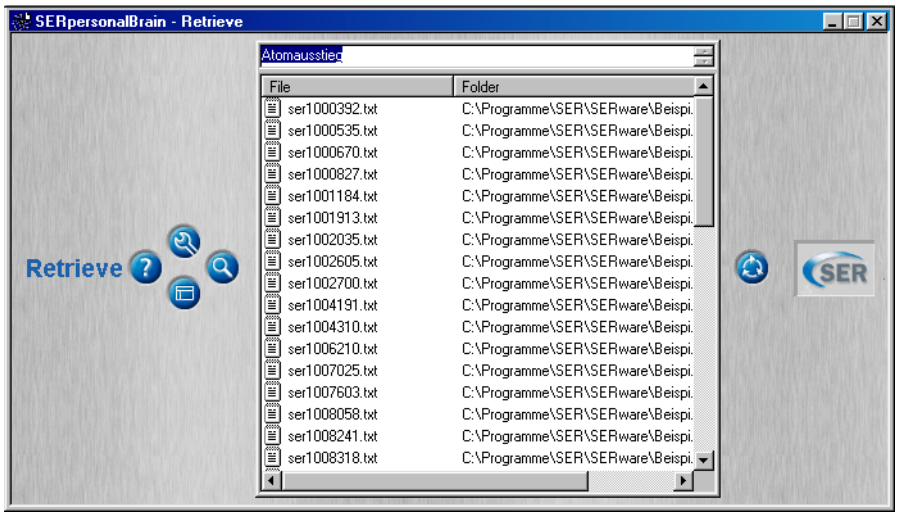


Abbildung 4.1: Das Hauptfenster von *SERPersonalBrain Retrieve*

ton ist und bestimmt, ob das interne oder das externe (die Standardanwendung für den jeweils ausgewählten Dateityp) Anzeigeprogramm verwendet werden soll. Mit der Funktion `Suche` wird die Suche gestartet. Die Suchbegriffe sind in dem Eingabefeld oben Mitte einzugeben. Die zur Suchanfrage passenden Dateien werden in der Tabellenliste darunter angezeigt. Die Liste ist unterteilt in die drei Spalten File (Dateiname), Folder (Verzeichnis) und - in der Abbildung nicht sichtbar - *Confidence*, einem Zahlenwert zwischen 0 und 100, der die Trefferquote widerspiegelt. Mit dem „Kreisverkehr"-Button rechts kann die Aufteilung des mittleren Bereichs verändert werden: Halb-halb (wie in der Abbildung), einzeiliges Eingabefeld (der Rest der Ergebnisliste) oder nur das Eingabefeld. Das daneben liegende Firmenlogo ist während des Suchvorgangs animiert. Die Suche ist asynchron, d.h. die Ergebnisliste füllt sich nach und nach und Einträge können noch während der Suche ausgewählt werden, um dann im Anzeigeprogramm dargestellt zu werden. Ein Unter- oder Abbrechen der Suche ist nicht möglich.
Die Anwendung verfügt noch über drei Kontextmenüs, die mit der rechten Maustaste angezeigt werden:
Das Kontextmenü 1 wird angezeigt, wenn der Anwender die rechte Maustaste irgendwo über der grauen Fläche oder den blauen Buttons oder die Tastenkombination Shift+F10 (Windows-Standard zur Anzeige eines Kontextmenüs per Tastatur) drückt, solange nicht das Eingabefeld den Focus besitzt. Das Kontextmenü 2 kann aufgerufen werden, wenn das Eingabefeld für die Suchbegriffe den Focus besitzt und Kontextmenü 3, wenn die Ergebnisliste den Focus innehat (auch hier

Kontextmenü 1:	Kontextmenü 2:	Kontextmenü 3:
Always on top	Rückgängig	Settings
Autohide	Ausschneiden	Details
About SERPersonalBrain	Kopieren	Start external application
Exit	Einfügen	
	Löschen	
	Alles markieren	

Tabelle 4.1: Die Kontextmenüs von *Retrieve*

muss die rechte Maustaste gedrückt werden).
Es fällt auf, dass das Kontextmenü 2 sowohl in deutsch vorliegt als auch Mnemonics (Zugriffstasten) für die einzelnen Menüoptionen bietet. Beides sind jedoch Standardeigenschaft eines Windows-Eingabefeldes und stehen somit auch automatisch zur Verfügung.
Zusammenfassend lässt sich die Benutzungsoberfläche als nicht intuitiv und wenig ergonomisch beurteilen. Aufgrund des geringen Funktionsumfanges der Anwendung ist diese jedoch schnell erlernt. Die Anwendung wurde mit der Methode der „Heuristischen Evaluation“ untersucht und mit dem *Windows-Styleguide* abgeglichen. Die genaue Untersuchung folgt später in diesem Kapitel.

Untersuchung der Funktionalität

Die *Retrieve*-Anwendung kann eine Vielzahl von Dokumenten durchsuchen. Dies sind z.B. einfache Text- oder HTML-Dateien, Word-Dokumente, aber auch Powerpoint-Präsentationen, Excel-Sheets oder PDF-Dokumente . Vor der ersten Suchanfrage müssen die potentiellen Dateien in das *Knowledge Repository* eingelesen werden. Hierzu muss der Anwender zunächst die zu durchsuchenden Verzeichnisse und Dateiendungen über den Konfigurationsdialog von *Retrieve* definieren.
Das *Knowledge Repository* mit den mitgelieferten Beispieldaten (572 dpa-Meldungen und Kochrezepte mit insgesamt 1,3 MByte Informationen) ist zügig aufgebaut (weniger als eine Minute). Eine einfache Suchanfrage, z.B. nach „Atomausstieg“, benötigt ca. elf Sekunden und liefert 46 passende Dateien. Erste Ergebnisse werden schon nach einer Sekunde angezeigt.
Zum Vergleich: Die Windows-Suche benötigt keine zwei Sekunden, um die gleichen 46 Dokumente zu finden. (Das Testsystem war ein mit einem 350 MHz Pentium II bestückter PC mit 128 MByte RAM).
Im Gegensatz zur Windows-Suche liefert *Retrieve* jedoch auch ähnliche Begriffe. Die Suche nach *Ausstieg* liefert mit *Retrieve* in 13 Sekunden 49 Dokumente, die Windows-Suche in weniger als drei Sekunden dagegen 56 Dokumente. Interessant

bei diesem Vergleich ist die Tatsache, dass *Retrieve* auch Dokumente mit dem Verb *stieg* (bei einer Confidence von 75 Punkten), nicht aber mit dem Begriff *Atomausstieg* gefunden hat.
Die Vergabe der Trefferquote beschränkt sich bei „Retrieve“ auf die Angaben „exakt gefunden“, entsprechend einer Confidence von 100 Punkten oder „ähnlichen Begriff gefunden“ mit einem geringeren Confidence-Wert. Es scheint *Retrieve* egal zu sein, wie oft der gesuchte Begriff gefunden wurde oder wo er im Text gestanden hat. Man könnte sich vorstellen, dass die *Confidence-Bewertung* höher ausfällt, wenn der gesuchte Begriff mehrmals im Dokument auftaucht oder weiter am Anfang als am Ende steht. Hier macht *Retrieve* jedoch keine genauere Unterscheidungen und vergibt in beiden Fällen 100 Punkte.
Die Stärke der Anwendung liegt jedoch nicht in der Suche nach einzelnen Begriffen, sondern in der Angabe von Beispielen, also Texten, die mit dem gesuchten Text inhaltlich ähnlich sind. Die besten Ergebnisse sollen sich mit Beispieltexten erzielen lassen, die zwischen einem Satz und einer Seite lang sind.
Um dies besser überprüfen zu können, wurde der eigene Dokumentenbestand in das Knowledge Repository aufgenommen. Hier fiel unangenehm auf, dass die Anwendung nur genau ein Repository unterstützt und StarOffice-Dokumente nicht verarbeiten kann. Außerdem ließen sich bestimmte Unterverzeichnisse nicht ausschließen.

Heuristische Evaluation von *Retrieve*

Für die Evaluation der Anwendung *SERPersonalBrain Retrieve* wurden die Heuristiken von Nielsen verwendet (siehe Seite 33 in dieser Arbeit).

Einfache und natürliche Dialoge

1. Die Platzierung der Buttons und Logos rechts und links der Tabelle beansprucht unnötig viel Platz, die man besser der Tabelle (deren Inhalt nicht vollständig zu sehen ist) gespendet hätte.

2. Die wichtige *Confidence*-Information steht in der Tabelle ganz rechts und ist daher meist unsichtbar. Besser: In der ersten Spalte.

3. Um die *Confidence*-Informationen sehen zu können, muss der Anwender horizontal scrollen. Das ist nicht einfach.

4. Die Bewertungszahlen für *Confidence* sind unklar. Was bedeutet ein Wert von 100? Heißt das Trefferquote 100%?

5. Eine gefundene Datei kann nur durch Doppelklick auf den Dateinamen angezeigt werden. Besser: Es sollte egal sein, wo der Anwender in der Zeile klickt.

6. Der Programmname *Retrieve* benötigt unnötig viel Platz und liefert keine wertvolle Information für den Anwender.

7. Auch das Logo benötigt unnötig viel Platz.

8. Auf der Registerkarte `Retrieve paths` kann eine Checkbox angeklickt werden („Use classification paths...“). Die Bedeutung ist unklar: Werden dann nur die Angaben der *Classify*-Anwendung berücksichtigt? Dann sollte das Listenfeld ausgegraut werden. Oder ergänzen die Angaben dort die hier gemachten Einstellungen? Dann sollte der Titel für die Checkbox geändert werden.

9. Die Bedeutung der Titel über den beiden Listen auf der Registerkarte `Extensions` ist unklar. Auf meinem System stehen mir wesentlich mehr Dateiendungen zur Verfügung als unter `available extensions` aufgeführt. Besser: `Supported Extensions`

10. Die Bedeutung der Abkürzung `Sub` auf der Registerkarte `Retrieve Paths` im Konfigurationsdialog ist unklar.

11. Die Bedeutung der Option `Case sensitive` im Highlights-Dialog von SERview ist unklar.

12. Die Bedeutung der Option `View 100%` im SERview-Kontextmenü ist unklar. Sind auch andere Werte als 100% möglich? Wenn ja, wie?

Sprich die Sprache des Benutzers

13. Die Benutzungsoberfläche ist komplett in englischer Sprache gehalten.

14. Auf der Registerkarte `Extensions` im `Retrieve Settings`-Dialog werden nur die technischen Endungen wie `.pdf`, `.doc` oder `.cdr` aufgeführt. Diese Endungen sind nur für erfahrende Anwender verständlich. Besser wären hier zusätzliche Klartextangaben wie z.B. *Word-Dokument*.

15. Der erste Reiter im Konfigurationsdialog ist mit `Update files` betitelt. Es werden jedoch nicht die Dateien, sondern das Repository aktualisiert. Besser: *Update Repository*.

16. Was meint die Menüoption `LDF Quality` im Kontextmenü von SERview?

Reduziere die Gedächtnisbelastung

Keine Benutzbarkeitsprobleme gefunden.

Sei konsistent

17. Trotz des gleichen Aussehens haben die vier Buttons links unterschiedliche Funktionen: Zwei öffnen ein neues Fenster, ein Button hat eine Umschalt-Funktion und einer führt eine Funktion direkt aus.

18. Das Kontextmenü 3 lässt sich nicht über `Shift+F10` anzeigen.

19. Das Kontextmenü 2 ist im Gegensatz zum Rest der Oberfläche deutschsprachig.

20. Einige Programmoptionen sind über den Konfigurationsdialog einzustellen, andere (nur) über die Kontextmenüs.

21. Das moderne Layout wird nicht über die ganze Anwendung hinweg eingehalten, z.B. nicht im Konfigurationsdialog.

22. Die OCR-Einstellungen finden sich auf der Registerkarte `Extensions`. Verbesserungsvorschlag: Eigene Registerkarte.

23. Die Benennung der Menüpunkte entspricht nicht den Konventionen. Hier ist die „Titel“-Schreibweise anzuwenden (z.B. „Start External Application“ statt „Start external application“).

24. Die Option `Details` ist nur über das Kontextmenü 3 ausführbar.

25. Das Kontextmenü von SERview führt Shortcuts auf.

26. Das Kontextmenü von SERview hat eine Reihe von Optionen, die nur exklusiv sind (z.B. Drucken).

27. Der Shortcut der Menüoption `Find Previous` in SERview ist `Ctrl+P`. `Ctrl+P` ist jedoch exklusiv der Funktion `Drucken` zuzuordnen.

Gib Feedback

28. Beim Start der Anwendung ist diese für einige Sekunden blockiert. Das animierte Logo zeigt dieses an, jedoch ist für den Anwender nicht ersichtlich, was die Anwendung gerade macht.

29. Es gibt keine Rückmeldung, wenn die Suche erfolglos war.

30. Es wird nicht angezeigt, wie viele Dokumente gefunden wurden.

31. Es ist nicht ersichtlich, nach welchem Kriterium (File, Folder, Confidence) die Liste sortiert ist.

32. Es ist nicht ersichtlich, ob die Liste auf- oder absteigend sortiert ist.

33. Auf der Registerkarte `Retrieve paths` ist der Button zum Neuanlegen eines weiteren Verzeichnisses immer ausgegraut.

34. Die beiden Symbolbuttons auf der Registerkarte `Retrieve paths` haben keine Tooltipps.

35. Die beiden Symbolbuttons auf der Registerkarte `Extensions` haben keine Tooltipps.

36. Die Buttons (First/Previous/Next/Last hit und Zoom in/out) im SERview-Fenster sind nicht deaktiviert, wenn sie keine Funktion haben (wenn also z.B. auf dem letzten Treffer positioniert ist oder sich das Dokument nicht weiter vergrößern lässt).

Klar gekennzeichnete Ausgänge

37. Es gibt (abgesehen vom [X]-Button) keinen sichtbaren Hinweis darauf, wie das Programm zu beenden ist.

38. Der Suchvorgang lässt sich nicht unterbrechen.

39. Die Aktualisierung des *Knowledge Repository* lässt sich nicht unterbrechen. Es ist auch nicht klar, wie lange der Vorgang voraussichtlich noch dauern wird.

Shortcuts

40. Die Bedienung nur über die Tastatur wird im Hauptfenster nicht unterstützt.

41. Die Bedienung über die Tastatur beschränkt sich im Konfigurationsdialog nur auf die Unterstützung der TAB-Taste. Mnemonics sind nicht vergeben worden.

Gute Fehlermeldungen

Keine Benutzbarkeitsprobleme gefunden.

Vermeide Fehler

42. Der Einstellungsdialog erlaubt auch auf Windows 9x-Systemen die Auswahl von PDF-Dokumenten. Diese werden jedoch nicht unterstützt. Besser: Nicht unterstützte Dokumenttypen nicht anzeigen oder ausgrauen.

43. Die Funktionsweise des Highlights-Konfigurationsdialogs in SERview ist sehr fehlerträchtig in Bezug auf die Farbauswahl.

44. Änderungen am Highlights-Farbschema werden nur aktiv, wenn zuvor die Checkbox *Refresh on OK* angeklickt wurde. Diese ist mit jedem neuen Öffnen des Dialogs wieder deaktiviert.

Hilfe und Dokumentation

45. Die Hilfe lässt sich nicht über F1 aufrufen.

46. Die Hilfe steht im Konfigurationsdialog überhaupt nicht zur Verfügung.

4.1.2 Classify

Die Teilanwendung *Classify* von *SERPersonalBrain* dient zur automatischen, wissensbasierten Klassifizierung von Dokumenten auf der lokalen Festplatte des Anwenders. Nachdem der Anwender das System „angelernt“ hat, soll es selbständig in der Lage sein, neue Dokumente in die selbstdefinierten Dokumentenkategorien einordnen zu können. So soll man ständig auf alle seine Dokumente schnell zugreifen können. Die Abbildung 4.2 zeigt das Hauptfenster der Anwendung.

Untersuchung der Benutzungsoberfläche

Die Anwendung *Classify* hat das gleiche „Look & Feel“ wie das oben besprochene Programm *Retrieve*. Die vier runden Schaltflächen aus dem Steuerungsbereich unten haben teilweise die gleiche Funktionalität wie die des Schwesterprogramms; das zuvor Gesagte trifft hier also auch zu. Es fällt auf, dass sich die rautenförmige Anordnung der Schaltflächen geändert hat.
Den größten Teil des Fensters nimmt diesmal der View-Bereich mit seinen Klassen (= Kategorien) ein. Eine Klasse wird durch ein Symbol und einen Namen

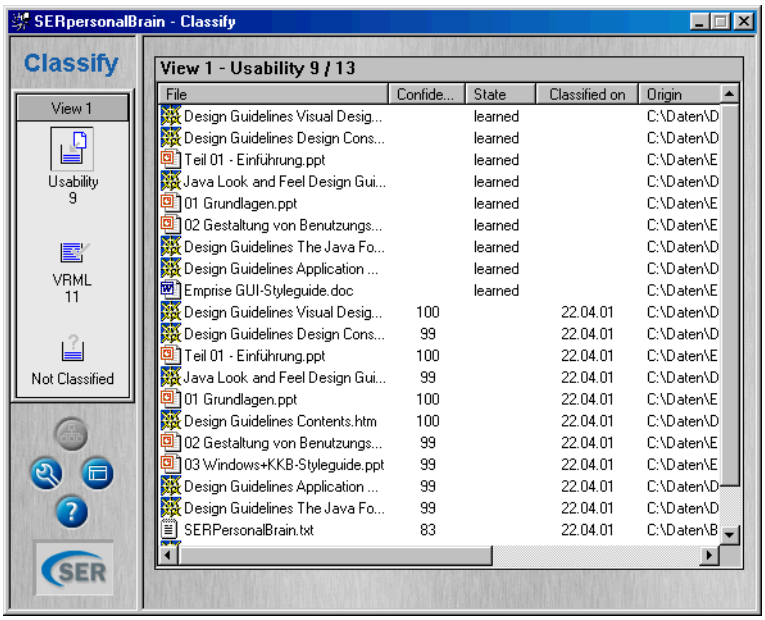


Abbildung 4.2: Das Hauptfenster von *SERPersonalBrain Classify*

repräsentiert. Das Anlegen erfolgt über eine entsprechende Menüoption im Kontextmenü (siehe auch unten). Der Anwender hat die Möglichkeit, unterschiedliche Sichten (Views) auf seine Dokumente anzulegen. Ein Dokument kann so in zwei oder mehr Klassen kategorisiert werden, vorausgesetzt, die Klassen befinden sich in disjunkten Views.
Beispiel: Möchte der Anwender seine Rezeptsammlung kategorisieren, so kann er diese z.B. in die folgenden Klassen einordnen:

- Vorspeisen,
- Hauptspeisen,
- Nachspeisen.

Ein Gericht wie „Melone mit Parmaschinken“ würde dann in die Klasse Vorspeise fallen, „Hähnchen mit Pommes“ dagegen in die Klasse Hauptgericht, genauso wie „Schollenfilet mit Kartoffelsalat“.
Eine andere Sicht auf Gerichte wäre dagegen

- Fischgerichte,

- Fleischgerichte,
- Vegetarische Gerichte.

Bei dieser Klassifizierung würden die Gerichte „Melone mit Parmaschinken“ und „Hähnchen mit Pommes“ in die Klasse Fleischgerichte fallen und das Schollenfilet in die Klasse Fischgerichte.
Die Anwendung gibt keinen sichtbaren Hinweis darauf, wie nun Dokumente den Klassen zugeordnet werden (oder dass das Programm vorher erst eine Lernphase benötigt). Erst nach dem Studium der Online-Hilfe ist klar, dass die Dokumente per *Drag and Drop* auf das jeweilige Klassensymbol gezogen werden müssen. Für jede Klasse werden mindestens zehn Dokumente benötigt, anhand derer die Anwendung die selbständige Klassifizierung „erlernt“. Sind nicht mindestens zehn Dokumente jeder Klasse zugeordnet, steht die automatische Klassifizierungfunktion nicht zur Verfügung. Dies könnte beim Anwender zu Problemen führen, wenn er z.B. eine neue Klasse einrichten möchte, für diese aber noch nicht genügend Dokumente hat. Eine Aktualisierung der Klassifizierung lässt sich nun nämlich nicht durchführen.
Der Platz wurde durch die hochkantige Anordnung der Elemente diesmal wesentlich effizienter ausgenutzt, jedoch benötigen der Anwendungsname und das Firmenlogo auch hier wieder viel ungenutzten Raum. Die Funktionalität der Anwendung wurde wieder in Kontextmenüs versteckt (siehe Tabelle 4.2). Durch einen „Einfachklick“ auf eines der Klassensymbole wird das Fenster erweitert. Ein erneuter Einfachklick auf ein Klassensymbol schließt diesen Erweiterungsteil wieder.

Kontextmenü 1	Kontextmenü 2	Kontextmenü 3	Kontextmenü 4
Always on top	Create Class	Rename Class	Small Icons
Autohide	Learn	Remove Class	Large Icons
About SERPersonalbrain	Classify		*Remove from learn set*
Exit	Rename View		*Add to learn set*
	Remove View		*Properties*
	Settings		Start external application
	Create View		

Tabelle 4.2: Die Kontextmenüs in *Classify*

Das Kontextmenü 1 ist mit dem aus *Retrieve* identisch. Bei den anderen beiden Menüs fällt auf, dass alle Menüoptionen über Zugriffstasten verfügen und die Benennungsregeln des Windows- Styleguides eingehalten wurden. Die Option `Settings` ist aber auch diesmal im Kontextmenü 2 unerwartet und zudem liegt sie genau zwischen zwei Funktionen, die sich auf das View beziehen.

Ungewöhnlich in Aussehen und Handhabung ist auch das dynamische Kontextmenü 4. Die Optionen `Small Icons` und `Large Icons` sind komplementär. Die Option `Remove from learn set` steht nur zur Verfügung, wenn eine „erlernte“ Datei vorhanden ist. Die Optionen `Add to learn set` und `Properties` sind nur für klassifizierte (aber noch nicht gelernte) Dateien vorhanden. Weitere Ungereimtheiten sind im Abschnitt 4.1.2 aufgeführt.

Untersuchung der Funktionalität

Die Erfolge der Klassifizierung scheinen wesentlich bessere Ergebnisse als die Suche zu liefern. Zumindest landeten bei meinen Versuchen nur wenige Dokumente in einer falschen Klasse. Allerdings ist die Zahl der Dokumente, die nicht klassifiziert werden konnten, relativ hoch. Hier müssten dem System wahrscheinlich mehr und bessere Dokumente zur Erlernung der Klassifikationsschemata gegeben werden.

Heuristische Evaluation von *Classify*

Die Punkte und Funktionalitäten, die mit *Retrieve* identisch sind, werden in der folgenden Übersicht mit (D) gekennzeichnet. Die gefundenen Probleme, die sich auf das Anzeigeprogramm *SERView* beziehen, werden hier nicht wiederholt.

Einfache und natürliche Dialoge

1. Es ist aus der Benutzungsoberfläche nicht ersichtlich, wie Dokumente einer Klasse zugeordnet werden.

2. Dokumente können nur durch *Drag and Drop* auf das Klassensymbol hinzugefügt werden, nicht jedoch durch Ziehen in das zugeordnete Dokumentenfeld.

3. Die Schaltfläche `Classify` ist solange deaktiviert, bis nicht jede Klasse mindestens zehn Dokumente enthält. Verbesserungsvorschlag: Im Tooltipp eine Information liefern.

4. Der Anwendungstitel und das Firmenlogo benötigen viel Platz und liefern für die Anwendung keine nützlichen Informationen (D).

5. Das Kontextmenü 2 (siehe Tabelle 4.2) ist sehr breit - mit entsprechend viel Leerraum am rechten Rand. Leere Stellen in einem Dialog lassen den

Anwender aber leicht vermuten, dass dort Informationen nicht angezeigt werden.

6. Entsprechendes gilt für das Kontextmenü 3 (siehe Tabelle 4.2).

7. „Erlernte“ Dokumente sind nach dem Klassifizierungsvorgang doppelt in der Dokumentfeld-Liste aufgeführt (einmal mit und einmal ohne Status *learned*).

8. Es dürfte für den Anwender (für mich ist es jedenfalls so) ein ungelöstes Rätsel bleiben, warum als Lernmenge definierte Dokumente durch die Klassifizierung nicht einen Confidence-Wert von 100 (Maximalwert) erhalten.

9. Die Funktionen `Create Class` und `Create View` stehen nur über das Kontextmenü zur Verfügung. Dies widerspricht dem Windows-Styleguide. Besser: Die Fenster mit einer Menüleiste und einem Menü `Edit` o.ä. versehen.

10. Die Funktionen `Learn`, `Classify`, `Rename View` und `Remove View` stehen nur über das Kontextmenü zur Verfügung. Dies widerspricht dem Windows-Styleguide.

11. Die Menüoption `Settings` steht inmitten der `View`-Funktionen. Diese Gruppierung ist nicht einleuchtend.

12. Die Menüoption `Create View` wird durch eine Trennlinie von den anderen `View`-Funktionen getrennt. Diese Anordnung ist nicht nachvollziehbar.

13. Die Bewertungszahlen für *Confidence* sind unklar. Was bedeutet ein Wert von 100? Heißt das Trefferquote 100%? (D)

14. Noch unklarer und unerklärt ist die Bedeutung des Wertes *Confidence distance* im Properties-Dialog.

15. Das Gleiche gilt auch für die Werte *Absolute confidence value* und *Relative distance of confidence values* im Settings-Dialog.

16. Siehe Punkt 9 im vorigen Abschnitt.

17. Siehe Punkt 10 im vorigen Abschnitt.

Sprich die Sprache des Bentzers

18. Siehe Punkt 13 im vorigen Abschnitt.
19. Siehe Punkt 14 im vorigen Abschnitt.
20. *Class* ist ein eher technischer Ausdruck. Besser wäre vielleicht *Category*.

Reduziere die Gedächtnisbelastung

Keine Benutzbarkeitsprobleme gefunden.

Sei konsistent

21. Siehe Punkt 17 im vorigen Abschnitt.
22. Siehe Punkt 20 im vorigen Abschnitt.
23. Die Kontextmenüs 2 bis 4 lassen sich – im Widerspruch zu den Forderungen des Windows-Styleguides – nicht über `Shift+F10` aufrufen.
24. Die Benennung der Menüpunkte entspricht nicht den Konventionen des Windows-Styleguides. Hier ist die „Titel"-Schreibweise anzuwenden (z.B. *Start External Application* statt *Start external application*. (D)
25. Siehe Punkt 21 im vorigen Abschnitt.
26. Siehe Punkt 22 im vorigen Abschnitt.
27. Die Anordnung der Schaltflächen im Steuerungsbereich unterscheidet sich von der in der Anwendung *Retrieve*.
28. Alle Menüoptionen im Kontextmenü 1 und die Option `Properties` im Kontextmenü 4 haben keine Mnemonics.
29. Ein geöffnetes Kontextmenü wird automatisch geschlossen und das neue angezeigt, wenn der Anwender irgendwo im Fenster einen „Rechtsklick" ausführt. Dies trifft nur nicht für das Kontextmenü 1 zu. Dieses muss immer explizit wieder geschlossen werden, bevor ein neues Kontextmenü angezeigt wird.
30. Beim Löschen einer Klasse wird die Dialogbox aus Abbildung 4.3 angezeigt. Hier werden deutsche und englische Sprache in der Meldung und den Schaltflächen vermischt.

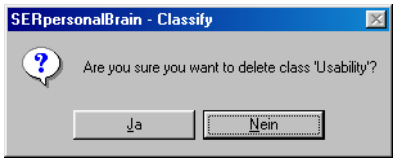


Abbildung 4.3: Meldungsdialog beim Löschen einer Klasse

31. Die Verwendung der Frage-Sprechblase als Symbol im Meldungsfenster aus Abbildung 4.3 verstößt gegen den Windows-Styleguide.

32. Die Verwendung zweier komplementärer Menüoptionen (`Small Icons` vs. `Large Icons` im Kontextmenü 4) ist für eine an sich binäre Eigenschaft ungewöhnlich.

33. Die Menüoptionen `Small Icons` und `Large Icons` entsprechen in ihrer Funktionalität der Menüoption `Details` des Kontextmenüs 4 aus der Anwendung *Retrieve*.

34. Wird das *Classify*-Hauptfenster so weit verkleinert, dass nicht alle Klassensymbole angezeigt werden können, werden die von den Scrollbalken bekannten Pfeil-Buttons angezeigt – der Scrollbalken jedoch nicht.

35. Im Tabellenkopf wird die Quelle einer Datei als *Origin* bezeichnet. In der Schwesteranwendung *Retrieve* wird dagegen die Bezeichnung *Folder* verwendet.

36. Im Properties-Dialog (siehe Abbildung 4.4) findet sich ein weiteres Synonym für Verzeichnis: *Path*. Unterschiedliche Benennungen der gleichen Sache können den Anwender verwirren und sollten vermieden werden.

37. Im Properties-Dialog wird ein Datum teilweise durch die Bezeichnung *Changed on* und teilweise durch *Classification date* eingeleitet (siehe auch nachfolgenden Punkt).

38. Im Properties-Dialog wird das Datum der Klassifizierung als *Classification date* bezeichnet; in der Tabelle dagegen als *Classified on* (vgl. vorangegangenen Punkt).

39. Die Menüoptionen `Learn` und `Classify` entsprechen (bei Hintereinanderausführung) der Schaltfläche `Classify`. Dies ist wegen der doppelten Namensgebung nicht konsistent.

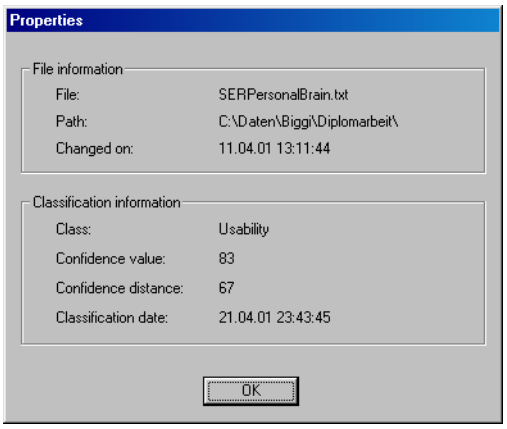


Abbildung 4.4: Properties-Dialog in *Classify*

40. Die Menüoptionen `Small Icons` und `Start external application` im Kontextmenü 4 haben die gleiche Zugriffstaste.

41. Die Schaltflächen im Konfigurationsdialog sind in Deutsch beschriftet, alle anderen Steuerelemente dagegen in Englisch.

Gib Feedback

42. Auf der Registerkarte *Classification paths* ist der Button zum Neuanlegen eines weiteren Verzeichnisses immer ausgegraut. (D)

43. Siehe Punkte 29 und 30 im vorigen Abschnitt.

44. Es dauert zu lange ($>$ 10 Sekunden) bis der Fortschrittsanzeigedialog der Klassifizierungsfunktion angezeigt wird.

45. Im Hauptfenster wird unterhalb des Klassensymbols angezeigt, wie viele „erlernte“ Dateien der Klasse zugeordnet sind, jedoch nicht – so wie im Dokumentenfeld – wie viele klassifizierte Dateien in dieser Klasse ebenfalls zu finden sind (siehe auch Punkt 55).

Klar gekennzeichnete Ausgänge

46. Siehe Punkt 32 im vorigen Abschnitt.

47. Der Klassifizierungsvorgang lässt sich nicht unter- oder abbrechen.

48. Das Dokumentenfeld kann nur durch Klick auf eines der Klassensymbole geschlossen werden. Besser: Im Dokumentenfeld-Teilfenster eine entsprechende Funktion anbieten.

49. Wird das Dokumentenfeld einer Klasse mit vielen Dokumenten geschlossen, so scheint zunächst die Liste neu aufgebaut zu werden (erkennbar an dem schrumpfenden Scrollbalken) bevor das Fenster tatsächlich geschlossen wird.

Shortcuts

50. Siehe Punkt 35 im vorigen Abschnitt.

51. Siehe Punkt 36 im vorigen Abschnitt.

52. Nicht alle Menüoptionen haben Mnemonics.

Gute Fehlermeldungen

53. Beim Zuordnen mehrerer Lerndokumente trat die Fehlermeldung aus Abbildung 4.5 auf. Diese technische Fehlermeldung ist für keinen Anwender verständlich.

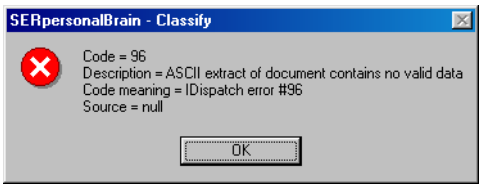


Abbildung 4.5: Fehlermeldung in *Classify*

Vermeide Fehler

54. Tritt der Fehler aus Abbildung 4.5 auf, so wird nach dessen Bestätigung die Aufnahme der restlichen, ausgewählten Dokumente in die Lernmenge abgebrochen.

55. Nach Beendigung des Klassifizierungsvorgangs wird der Anwender nicht darüber informiert, dass Dokumente in die Pseudo-Klasse *Not Classified* eingeordnet wurden (d.h. die Klassifizierung ist für diese Dokumente fehlgeschlagen).

56. Erlernte Dateien lassen sich ohne Nachfrage löschen.

57. Aufgrund der doppelten Auflistung von Dokumenten kann das selbe Dokument im Zustand *learned* und *not learned* sein.

Hilfe und Dokumentation

58. Siehe Punkt 40 im vorigen Abschnitt.

59. Siehe Punkt 41 im vorigen Abschnitt.

60. Die Online-Hilfe ist nicht kontext-sensitiv.

61. Das Programm und die Hilfe geben keine Auskunft darüber, welche Stati ein Dokument einnehmen kann.

4.2 Inxight Tree Studio

Mit dem Programm *Inxight Tree Studio* von der Firma `Inxight` lassen sich die in Abschnitt 6.1.6 vorgestellten Hyperbolic Trees erstellen. Da es eines der wenigen Programme ist, die eine der in Kapitel 6 vorgestellten Visualisierungstechniken kommerziell anbietet[2], habe ich dieses Programm einer Heuristischen Evaluation unterzogen. Grundlage dieser Evaluation ist die Version 1.5.

Die Anwendung besteht aus zwei Teilprogrammen: Mit dem Hauptprogrammm *Inxight Tree Studio* lassen sich die Datenquellen für die Erstellung der Hyperbolic Trees definieren. Diese können dann mit de zweiten Anwendung, dem *Star Tree Designer*, bearbeitet werden.
Zunächst betrachte ich das Hauptprogramm und anschließend den *Star Tree Designer*. Dabei wird zuerst eine generelle Einführung in die Programme gegeben und danach werden sie einer Heuristischen Evaluation unterzogen.

4.2.1 Inxight Tree Studio (Hauptprogramm)

Für die Erstellung eines neuen Projektes stehen dem Anwender fünf Möglichkeiten zur Verfügung:

- Ein leeres Projekt, dem dann von Hand Elemente hinzugefügt werden können.

- Ein Projekt, dessen Datengrundlage eine Website bildet. Wählt man diese Option aus, so müssen auf den folgenden Seiten einige Angaben gemacht werden, wie z.B. die Start-URL oder welche Links bis in welche Tiefe verfolgt werden sollen. Sind alle Angaben gemacht, wird die Website geladen und deren Struktur ermittelt.

- Eine Import-Datei im CSV-Format[3]. Auf den Folgeseiten lassen sich dann diverse Einstellungen dazu vornehmen, in welcher Spalte welche Informationen zum Aufbau des Baumes zu finden sind.

- Ein bestehender *Star Tree*.

[2]Eine dreissigtägige Testversion des Programms findet man unter http://www.inxight.com/products_wb/tree_studio/

[3]Ausgangspunkt einer CSV- Datei ist ein Tabellenblatt, z. B. aus dem Tabellenkalkulationsprogramm *Excel* von der Firma `Microsoft`. Die Werte einer Zeile sind beim CSV-Format durch Semikoli getrennt.

- Ein bestehendes Projekt.

Die Anwendung ist nicht lokalisiert, so dass man bei der Benutzung der Standard-Dialoge mit zwei Sprachen konfrontiert wird: Englischsprachige Bezeichner für alle programmspezifischen Steuerelemente und deutsche Bezeichner für die Standard-Schaltflächen wie *Abbrechen* oder *Hilfe*.
Die Anwendung erlaubt es zwar, mehrere Projekte gleichzeitig geöffnet zu haben, jedoch werden keine Transfer-Funktionen (wie *Cut*, *Copy* oder *Paste*) unterstützt. Daten aus anderen Projekten lassen sich nur mühsam übernehmen (über die Menüoption *Project* → *Copy into project* → *Object*).
In Abbildung 4.6 ist das Hauptfenster der Anwendung mit einer geöffneten Projektdatei zu sehen.

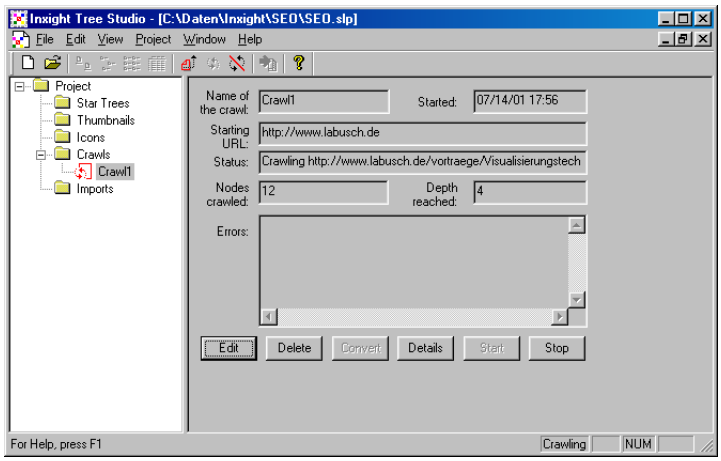


Abbildung 4.6: Das Hauptfenster von Inxight Tree Studio beim Einlesen der Struktur einer Website.

Ein fertig erstellter *Star Tree* kann beispielsweise als Java-Applet auf einer Website veröffentlicht werden. Das Applet kann dabei zunächst lokal oder direkt über eine FTP[4]-Verbindung auf dem Zielrechner veröffentlicht werden.
Ebenso wie der *Star Tree Designer* unterstützt die Anwendung den Benutzer durch eine aktive Hilfe, insbesondere bei „kritischen“ Operationen, da die Anwendung über keine *Undo*-Funktion verfügt.

[4]FTP = File Transfer Protocol ist ein Protokoll zur Übertragung von Dateien zwischen zwei Rechnern im Internet.

Heuristische Evaluation von *Inxight Tree Studio*

Einfache und natürliche Dialoge

1. Der im Fenster vorhandene Platz wird nicht ausgenutzt. Wie in Abbildung 4.6 zu sehen ist, müssen die Labels nicht zweizeilig sein (z.B. *Name of the Crawl*).

2. In der Statuszeile wird neben der aktiven Hilfe auch angezeigt, ob die Feststelltaste (Caps Lock) oder die *Num*-Taste aktiv ist. Diese Informationen sind überflüssig, da der Zustand dieser Tasten nur für die Anwendung an sich betrachtet keine Bedeutung hat.

3. Die Dialoge für Star Trees, Thumbnails oder Icons geben Informationen über die Größe der entsprechenden Datei wieder. Diese Information ist überflüssig, da ein erzeugtes Star Tree Applet für eine Website die Daten komprimiert bzw. weitere Dateien hinzufügt (z.B. die Archivdatei mit den Java-Class Dateien).

4. Der Einstelldialog für Star Trees enthält rechts ein leeres quadratisches Textanzeigesteuerelement, das nicht beschriftet ist. Die Bedeutung ist unklar. (Laut Hilfe sollen hier Informationen den Star Tree betreffend angezeigt werden. Dafür scheint das Steuerelement jedoch zu klein zu sein).

5. Im Steuerelement *Name of the Star Tree* wird nicht der Name des Trees, sondern der Dateiname angezeigt.

6. Die Anwendung erlaubt es, mehrere Projekte gleichzeitig offen zu haben, jedoch werden keine Transfer-Funktionen (*Cut, Copy, Paste*) unterstützt.

Sprich die Sprache des Benutzers

7. Bei allen Dialogen, die auf Standard-Dialogen von `MS Windows` aufbauen, sind die programmspezifischen Bezeichner in Englisch, die Bezeichner der Standard-Steuerelemente jedoch Deutsch (z.B. die Standard-Schaltflächen wie *Abbrechen*).

Reduziere die Gedächtnisbelastung

Keine Benutzbarkeitsprobleme gefunden.

Sei konsistent

8. Die Menüoption *Documentation Overview* aus dem Hilfe-Menü zeigt die Informationen im Gegensatz zu den anderen Hilfe-Optionen in einem Webbrowser an und benutzt nicht die `Windows`-Hilfe.

9. Die Einstelldialoge für Icons und Thumbnails sind zwar von den Steuerelementen her identisch, das Layout der Dialoge dagegen nicht.

Gib Feedback

10. Während eines *Crawl*-Prozesses gibt es keine Rückmeldungen darüber, wie viele Seiten noch geladen und nach wie vielen weiteren Verweisen noch gesucht werden müssen. Auch eine Restzeitanzeige fehlt hier.

Klar gekennzeichnete Ausgänge

Keine Benutzbarkeitsprobleme gefunden.

Shortcuts

Keine Benutzbarkeitsprobleme gefunden.

Gute Fehlermeldungen

11. Beim ersten Start stürzte das Programm ab und es wurde die Standardfehlermeldung für derartige Situationen angezeigt (siehe Abbildung 4.7). Die Ursache des Fehlers konnte dadurch dem Anwender nicht mitgeteilt werden, so dass das ungute Gefühl einer instabilen Anwendung besteht (bei einem Softwareprodukt, dessen Lizenz $5000 beträgt).

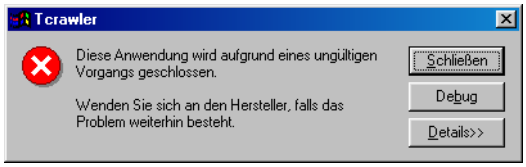


Abbildung 4.7: Die Anwendung stürzte beim ersten Anlauf ab.

12. Schlägt ein Crawl-Prozess (Ermitteln der Struktur einer Website aus dem Internet) fehl, so steht in der Protokolldatei u.a. die Fehlermeldung *Invalid*

Crawl Domain http://www.labusch.de/ cannot be reached. Die Adresse ist jedoch nicht falsch, sondern war aufgrund einer fehlerhaften Internetverbindung nicht erreichbar.

Vermeide Fehler

Keine Benutzbarkeitsprobleme gefunden.

Hilfe und Dokumentation

Keine Benutzbarkeitsprobleme gefunden.

4.2.2 Star Tree Designer

Mit der Anwendung *Star Tree Designer* lassen sich die mit dem *Tree Studio* erstellten Bäume bearbeiten. Man kann neue Knoten hinzufügen, bestehende löschen oder verschieben und an andere Zweige hängen. Für jeden Knoten kann man eigene Farben für Vorder- und Hintergrund vergehen oder diese mit einem Icon oder Thumbnail (kleines Vorschaubild) versehen. In Abbildung 4.8 ist die Anwendung zu sehen.
Um die Übersicht zu behalten, lassen sich einzelne Zweige des Baums ein- und ausblenden. Diese Funktion wird über ein kleines Steuerelement unten rechts in einem Knoten ausgelöst.
Korrespondiert ein Knoten mit einer Webseite im Internet, so wird dessen Adresse in der Statuszeile angezeigt. Vermutlich lassen sich diese Verknüpfungen auch von Hand anlegen (es gibt eine Funktion namens *Create Link*, jedoch ist diese Funktion oft inaktiv und es wird nicht ganz klar, warum dies so ist.
Weitere Funktionen der Anwendung betreffen die Darstellung des Baumes als Ganzes: Man kann bestimmen, wie sich der Baum aufspannen soll, wie groß die Abstände zwischen den einzelnen Zweigen sein sollen und welche Informationen zu einem Knoten angezeigt werden sollen und wo zu positionieren sind (z.B. ober- oder unterhalb der Beschriftung).

Heuristische Evaluation von *Star Tree Designer*

Einfache und natürliche Dialoge

1. Das Dialogfenster für die Knotenattribute hat nur die drei Schaltflächen Reset, Apply und Cancel (sic! sie sind nicht deutsch). Oft möchte man nur

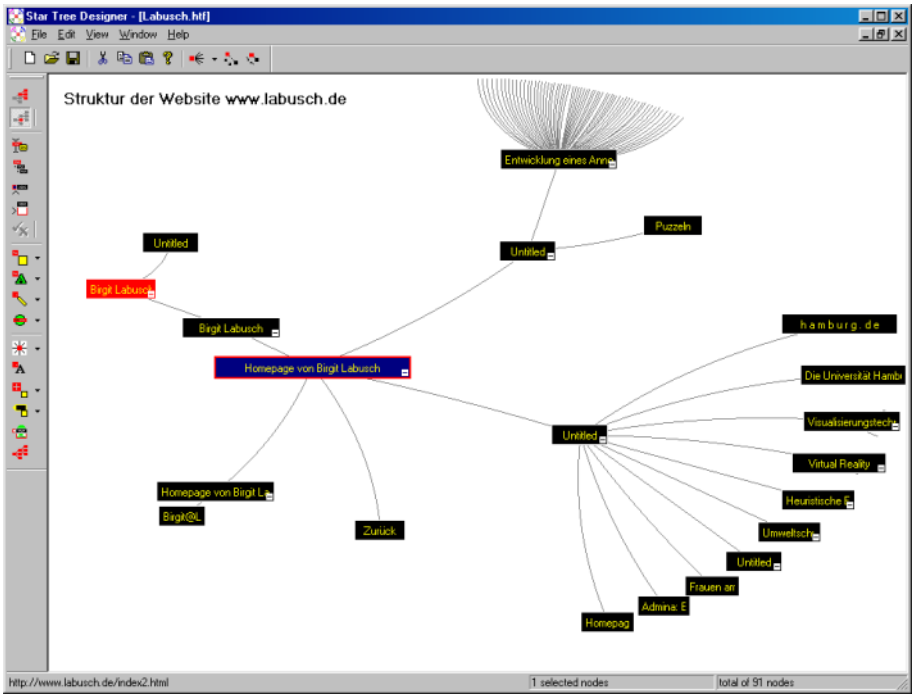


Abbildung 4.8: Ein mit Inxight Tree Studio erstellter Graph

schnell den Namen ändern. Um den Dialog dann zu verlassen, muss man zunächst auf Apply und dann auf Cancel klicken.

2. Neben den Standard-Symbolen zum Öffnen oder Speichern von Dateien bringt der Star Tree Designer auch eine Reihe eigener Symbole mit, die hauptsächlich aus den Farben Schwarz, Rot, Gelb und Grün aufgebaut sind und sich wenig voneinander unterscheiden (siehe Abbildung 4.9).

Abbildung 4.9: Auswahl einiger Symbole des Star Tree Designers (siehe auch Farbtafel 1)

3. Man kann für einen Knoten eine „Quantity Color“ vergeben. Was es damit auf sich hat und wozu es gut ist, bleibt unklar.

4. Die Symbolleiste enthält einige Funktionen (z.B. Increase Spacing), die exklusiv sind. Dies Widerspricht auch dem Windows- Styleguide.

5. Klickt der Anwender unmotiviert irgendwo in das Fenster mit dem Star Tree, so wird seine Position nach diesem Punkt neu ausgerichtet. Wie das sein wird, ist weder vorherzusehen, noch erwartet man, daß bei einem einfachen Klick irgendwo hin irgendwas passiert.

6. Knoten lassen sich per durch *Drag*-Operationen mit der Maus verschieben. Dabei passt sich auch der ganze Baum mit an, was zumindest für den Anfänger nicht nachvollziehbare Auswirkungen hat.

7. Fast alle Funktionen des Programms sind auch über Symbole in Symbolleisten erreichbar, die standardmäßig alle eingeblendet sind. Die Symbole sind sich zum Teil sehr ähnlich und erschweren es so dem Benutzer, die Symbole wiederzuerkennen.

Sprich die Sprache des Benutzers

8. Die Anwendung und alle Bezeichner sind komplett in Englisch gehalten. Einzige Ausnahme: Die Sicherheitsabfrage beim Schliessen der Anwendung und noch nicht gesicherter Änderungen: Dieser Dialog ist komplett Deutsch.

Reduziere die Gedächtnisbelastung

9. Im Dialog aus Abbildung 4.10 ist die Bedeutung der Symbole ohne Tooltipps zunächst unklar; danach immer noch verwirrend. Die letzten beiden Symbole dienen der Knoten-Navigation durch Geschwisterknoten. Mit dem letzen Symbol gelangt man aber nicht zum nächsten linken Geschwisterknoten, sondern – wie der Tooltipp sagt – zum vorherigen. Und der ist rechts.

10. Im Dialog „Node Display Options“ (siehe Abbildung 4.11) können über Symbole einige visuelle Einstellungen vorgenommen werden. Die Bedeutung der Symbole ist wenig intuitiv.

Sei konsistent

11. Die Info-Menüoption im Hilfe-Menü wird im Widerspruch zum Windows-Styleguide mit einer Ellipse (...) abgeschlossen.

Abbildung 4.10: Beschriftungsdialog mit unklaren Symbolen

12. Die Symbole acht bis zwölf (siehe Abbildung 4.9) dienen der Farbmanipulation. Die Bedeutung Trennstrich zwischen dem elften (Quantity Color) und zwöften Symbol (Background Color) ist unklar.[5]

Gib Feedback

13. Erstellt man den Star Tree von Hand und fügt nach und nach Knoten hinzu, kann es passieren, dass der neue Knoten nicht mehr im sichtbaren Bereich liegt. Der Anwender muss dann zuerst die Position des Baumes anpassen, bevor der neue Knoten bearbeitet werden kann.

14. Wählt man im Mehrfachauswahlmodus einen weiteren Knoten aus, so vergeht etwas mehr als eine Sekunde, bis die Auswahl visuell zurückgemeldet wird.

15. Die Funktion *Create Link* ist oftmals ohne ersichtlichen Grund inaktiv.

Klar gekennzeichnete Ausgänge

16. Das Fenster zur Einstellung der Attribute eines Knotens verfügt nur über einen Cancel- und den X-Button. Nach Übernahme der Änderungen muss der Dialog über diesen Weg verlassen werden. Dieser Ausgang ist weder klar, noch vertrauensschaffend.

[5] Bei einer anderen Sichtweise wird die Bedeutung des Trennstrichs klar: Die Symbole links vom Trennstrich beziehen sich auf Funktionen, die einen Knoten manipulieren, die rechts vom Trennstrich betreffen den gesamten Baum.

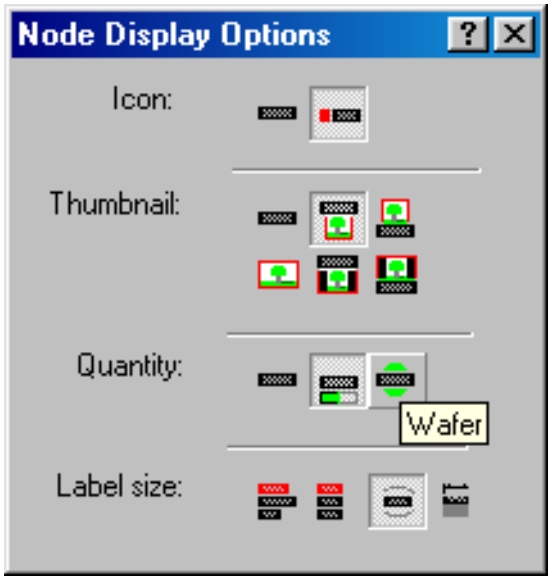


Abbildung 4.11: Dialog „Node Display Options“

17. Kleine Dialoge (wie z.B. in Abbildung 4.11) können nur über den X-Button beendet werden.

Shortcuts

16. Die Attributeigenschaften eines Knotens lassen sich entweder über das Menü Bearbeiten, das Kontextmenü oder per Doppelklick aufrufen. Wird der Doppelklick jedoch nicht schnell genug ausgeführt, werden die Klicks als Positionsänderungsbefehle interpretiert. Nach dem ersten Klick verschiebt sich also der Knoten und der zweite landet im Leeren.

Gute Fehlermeldungen

17. Wendet man die Funktion *Create Link* auf einen bereits verlinkten Knoten an, wir die Fehlermeldung *There is already a link between the specified nodes.* angezeigt. Es bleibt jedoch unklar, welcher der zweite Knoten sein könnte.

Vermeide Fehler

18. Die Anwendung verfügt über keine Undo-Funktion. Bei Löschoperationen wird man jedoch gewarnt, sofern man die Dialogbox nicht ausgeschaltet hat.

Hilfe und Dokumentation

Keine Benutzbarkeitsprobleme gefunden.

4.3 Mind Manager

Das Programm *MindManager* der Firma `Mindjet` ermöglicht die interaktive Erstellung sog. Mind Maps (siehe z.B. [Buzan et al. 1997]).
Mind Maps wurden ursprünglich von Tony Buzan zur Ausarbeitung, Beschreibung und Kommunikation komplexer Situationen und Ideen und als Lernhilfe entwickelt. Eine Mind Map soll dabei beide Gehirnhälften ansprechen: Die rational-logische linke Hälfte und die auf optische und kreative Reize reagierende rechte Hälfte. Dies, so die Theorie hinter den Mind Maps, führt zu einem schnelleren und tieferen Einblick in ein Thema.
Mind Maps lassen sich auch als Wissenskarten oder genauer als Wissensstrukturkarten im Bereich des Wissensmanagements einsetzen (siehe hierzu Kapitel 2.4). Abbildung 4.12 zeigt eine solche Karte im Hauptfenster der Anwendung.
Die Anwendung *MindManager* ist eine Standard `MS Windows`-MDI-Anwendung[6]. Einem mit `MS Windows` vertrauten Anwender bereitet die Handhabung dieser Oberfläche daher keine Probleme.
Ungewöhnlich für eine MDI-Anwendung ist der untere Teil des Fensters. Funktion und Zweck dieser weißen Fläche ist zunächst nicht klar. Es handelt sich hierbei um das „angedockte“[7] Notizfenster. Zu einem Zweig in der Mind Map kann der Anwender hier Notizen erfassen.
Ist der Anwender mit Mind Maps vertraut, so kann er sehr schnell erste Erfolge verzeichnen, da sich das Programm sehr intuitiv bedienen lässt. Dies gilt insbesondere für die *Drag and Drop*-Möglichkeiten. Dies ist für ein so komplexes

[6] MDI = Multiple Document Interface

[7] In modernen Windows- Anwendungen können die Symbolleisten beliebig platziert werden. Es ist auch möglich, sie als eigene Fenster (sog. Palletten-Fenster) auf der Oberfläche zu halten. Dies ist über die Steuerelemente an der linken Seite der Symbolleiste möglich. Eine Symbolleiste muss nicht zwangsweise nur Symbolbuttons enthalten. Hier können alle beliebigen Steuerelemente angeordnet werden. Ist die Symbolleiste Teil des Hauptfensters, so sagt man auch, sie sei „angedockt“.

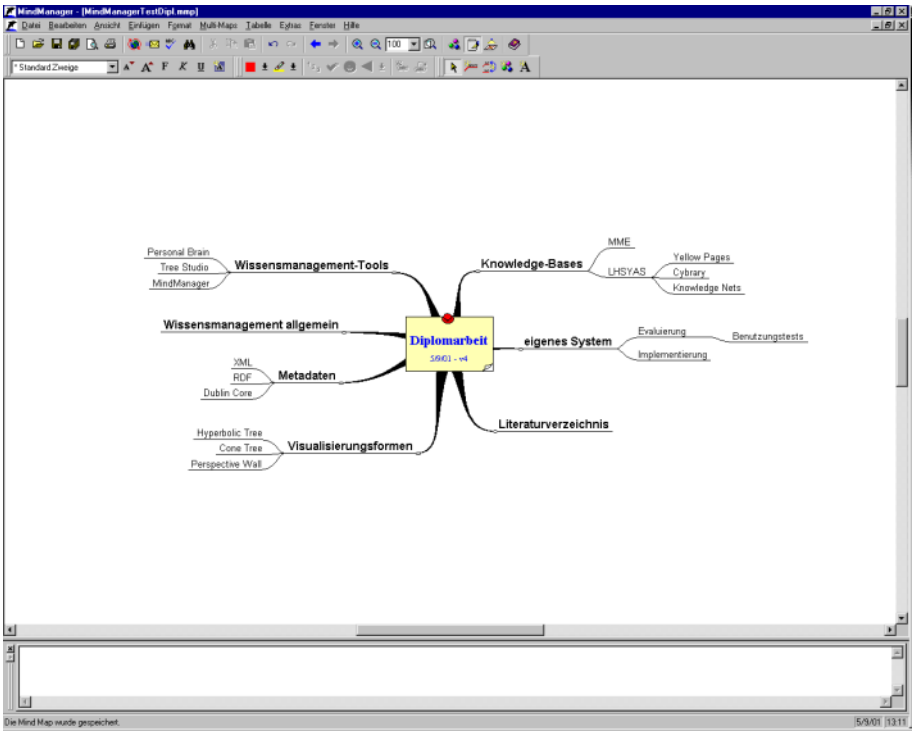


Abbildung 4.12: Das Hauptfenster von MindManager mit einer MindMap.

Programm wie dieses positiv hervorzuheben. Der schnelle Lernerfolg stellt sich auch dadurch ein, dass das Programm die Exploration neuer Funktionen aktiv unterstützt. Wird zum Beispiel das erste Mal die Menüoption *Conference* aufgerufen, so wird automatisch die Online- Hilfe zu diesem Thema mit einem kurzen Tutorial aufgerufen. Bei weiteren Aufrufen unterbleibt aber diese Aktion.

Das Programm versucht den Anwender auch von so vielen technischen Eigenschaften wie möglich abzuhalten. Für die Conference-Funktion ist beispielsweise ein Zugang zum Internet notwendig. Das Programm bietet dem Anwender eine Testfunktion, um die Anbindung an das Internet zu überprüfen (siehe Abbildung 4.13). Falls hierfür der Weg über eine Firewall notwendig ist bzw. ein Proxyserver verwendet wird, erkennt das Programm dies automatisch.

Bei der Evaluation der Anwendung wurde hauptsächlich der Teil zur Erstellung von Mind- bzw. Knowledge-Maps betrachtet. Die Programmbesonderheiten wie diverse Format- und Exportfunktionen, Multimaps oder die Möglichkeit der gruppenbasierten Bearbeitung von Mind Maps über das Internet wurden nicht betrachtet. Das Ergebnis der Evaluation ist sehr positiv, was sich auch in der

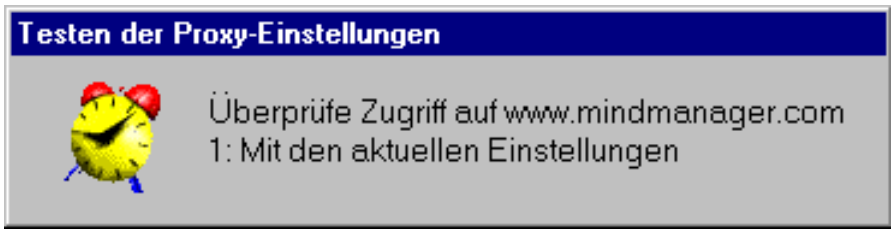


Abbildung 4.13: Das Hauptfenster von MindManager mit der Internet-Anbindung.

geringen Anzahl der gefundenen Usability- Probleme zeigt (unter der Berücksichtigung, dass ein Anwender einerseits mit der Bedienung des Betriebssystems `MS Windows` vertraut ist und andererseits ihm der Umgang mit Mind Maps bekannt ist).
Die Anwendung enthält viele innovative Ideen, die auch benutzungsfreundlich umgesetzt wurden. So gibt es z.B. im Hilfe-Menü die Option *Info Center*. Das Info-Center ergänzt die Hilfe-Funktionen um aktuelle Informationen. Hierbei kann sich z.B. um eine Liste der häufig gestellten Fragen (FAQs) oder um aktuelle Programmänderungen handeln. Diese Textdateien können dann bei einer bestehenden Internetverbindung einfach per Knopfdruck aktualisiert werden.

Heuristische Evaluation des *Mind Managers*

Einfache und natürliche Dialoge

1. Die Symbole entsprechen zum Großteil dem Windows-Standard, jedoch sind die spezifischen Symbole des Programms weniger aussagekräftig. Ihre Farbvielheit lässt die Symbolleisten unruhig wirken.

2. In der Statuszeile wird das aktuelle Datum und die Uhrzeit angezeigt. Diese Information ist überflüssig (die gesamte Statuszeile lässt sich jedoch ausblenden).

Sprich die Sprache des Benutzers

3. War der oben beschriebene Internet-Zugriffs-Test erfolgreich, so wird die Meldung aus Abbildung 4.14 angezeigt. Hier passt das Symbol (Warnung) nicht zur Aussage (Information) der Meldung.

4. Die eingebaute Rechtschreibprüfung kennt nicht den Begriff „Mind Map“.

Abbildung 4.14: Aussage und Darstellung dieser Meldung stimmen nicht überein.

Reduziere die Gedächtnisbelastung

Keine Benutzbarkeitsprobleme gefunden.

Sei konsistent

5. Der untere Teil des Hauptfensters ist bei MDI-Anwendungen nicht üblich.

Gib Feedback

6. Die Bedeutung des unteren Teils des Haupftfensters ist unklar, da hier auf jegliche Beschriftungen verzichtet wurde. Für eine MDI- Anwendung ist dieses Layout auch ungewöhnlich.

7. Wenn kein Zweig in der MindMap ausgewählt ist, kann im Notizbereich keine Eingabe erfolgen. Wird der Versuch aber trotzdem unternommen, so gibt es weder ein visuelles noch ein akustisches Feedback.

8. Das Notizfenster (wenn es als Palletten-Fenster dargestellt wird) gibt nicht die volle Bezeichnung des ausgewählten Zweigs wieder. Dies geschieht auch dann, wenn genügend Platz in der Titelleiste des Fensters vorhanden wäre.

Klar gekennzeichnete Ausgänge

Keine Benutzbarkeitsprobleme gefunden.

Shortcuts

Keine Benutzbarkeitsprobleme gefunden.

Gute Fehlermeldungen

Keine Benutzbarkeitsprobleme gefunden.

Vermeide Fehler

9. Hat eine Menüoption Untermenüpunkte und sind in diesem alle Optionen deaktiviert, so ist die übergeordnete Menüoption aktiviert. Dies kann den Anwender fälschlicherweise vermuten lassen, dass in dem Untermenü auswählbare Optionen vorhanden sind.

10. Die Symbolbuttons zur Vergabe der Vorder- und Hintergrundfarbe können jederzeit ausgewählt werden, haben jedoch keine Wirkung, wenn nicht ein Element aus der Mind Map zuvor ausgewählt wurde.

Hilfe und Dokumentation

Keine Benutzbarkeitsprobleme gefunden.

Kapitel 5

Metadaten und Wissensmanagement

Eine Aufgabe des Wissensmanagements ist das Wiederauffinden von Informationen oder Wissen. Metadaten helfen einerseits, diese Informationen zu speichern und andererseits, um geeignete Visualisierungsformen zu finden. Ziel dieses Kapitels ist es, nach einer Begriffsbegriffsbestimmung einige Metadatenkonzepte zu beschreiben und zu untersuchen, wie die Metadaten verwendet werden.
Neben diesen offensichtlichen Metadatenkonzepten kann man auch Datenstrukturen als Metadaten auffassen. Aus diesem Grund wird im zweiten Teil untersucht, welche Metadaten sich in Datenstrukturen wiederfinden lassen. Diese Metadaten werden die Art und Weise beeinflussen, wie der von mir implementierte Prototyp aussehen wird.
Viele verschiedene Assoziationen entstehen bei unterschiedlichen Menschen, wenn sie den Begriff „Metadaten“ hören. Eine sehr allgemeine Begriffsbeschreibung definiert sie als „Daten über Daten“. Eine kurze und prägnante Definition liefert die ISO Spezifikation 11179 [ISO 1995]. Sie beschreibt Metadaten als:

> „The information and documentation which makes data sets understandable and shareable for users.“

Es stellt sich somit die Frage: „Wofür sind Metadaten sinnvoll?“
Zum Einen ist dies bei der Suche in großen Datenbeständen der Fall. Statt große Datenbestände zu durchsuchen, ist es effektiver, Metadaten über diese Bestände anzulegen und diese dann zu durchsuchen, da sie um ein vielfaches kleiner und besser strukturiert sind, was die Suche erheblich verkürzt.
Zum Anderen bei der Datendokumentation, denn die Daten, die sich in den letzten Jahrzehnten in den Firmen angesammelt haben, sind nahezu völlig undokumentiert. Sind auch die Mitarbeiter nicht mehr in dem Betrieb beschäfigt, so

werden die Daten völlig wertlos. Diesem Mißstand wird schon durch die sog. „Data Mining-Werkzeuge“ begegnet, die das Durchsuchen von undokumentierten Datenbeständen ermöglichen sollen. Eine langfristige Werterhaltung von Daten lässt sich aber am ehesten durch die Verwendung von Metadaten erzielen. Außerdem kann man in gewissem Maße Datenredundanz vermeiden und mit Hilfe der Metadaten etwas über die Datenqualität aussagen. Zusätzlich kann noch eine Art „Haltbarkeitsdatum“ eingeführt werden, das die Aktualität der angezeigten Daten sichert.
Metadaten können inhaltsab- oder inhaltsunabhängig sein. Zu den inhaltsabhängigen gehören solche Daten, die zwar den Inhalt eines Dokumentes beschreiben (semantische Metadaten), aber nicht explizit in dem Dokument enthalten sind. Metadaten, die zum Interpretieren der Daten geeignet sind, sind ebenfalls inhaltsabhängig. Sie geben Auskunft über den syntaktischen Aufbau der Daten (z.B. Informationen über die Datentypen der Datenfelder, den Aufbau von Datenfeldern, die Pixelanzahl eines Bildes). Metadaten, die weder syntaktischer noch semantischer Natur sind, aber trotzdem inhaltsabhängig, werden als Inhaltsmetadaten bezeichnet (z.B. Schlüsselwörter, Farben eines Bildes). Mit diesen inhaltabhängigen Metadaten beschäftigt sich Abschnitt 5.2.
Zu den inhaltsunabhängigen Metadaten gehören solche Metadaten, die die zugehörigen Daten eindeutig identifizieren, wie z.B. den Namen des Autors und der des Dokuments, eine Versionsnummer, das Datum des letzten Zugriffs. Ebenfalls inhaltsunabhängig sind administrative Metadaten. Sie sollen die Verwaltung von Daten erleichtern (z.B. Aufbewahrungsort von bestimmten Dateien (welches Verzeichnis auf welcher Festplatte, eine URL) oder Beziehungen zu anderen Dateien/Daten). Auf diese Art der Metadaten wird im folgenden eingegangen.

5.1 Beschreibungssprachen für Metadaten

In den letzten Jahren haben sich verschiedene Metadaten-Sprachen entwickelt. Einige von ihnen sollen im folgenden beschrieben werden:

5.1.1 Dublin Core

Dublin Core (DC) wurde mit dem Ziel entwickelt, neben den bekannten Bibliotheksdaten weitere Informationen über ein Dokument speichern zu können, die der Autor selbst bestimmen kann. Ein weiterer sehr wichtiger Aspekt bei der Entwicklung des DC war das einfache Auffinden von Dokumenten, die in einem Netzwerk (wie z. B. dem Internet) abgelegt sind.

Wissenschaftler nutzen Dublin Core Metadaten im Internet einerseits zur besseren Verfügbarkeit ihrer Arbeitsergebnisse, andererseits zur besseren Darstellung ihrer Arbeitsgruppen, Lehrveranstaltungen und Forschungsprojekte. Die Metadaten erlauben ein hochwertiges Retrieval auf diesem spezifischen Material.
Es folgt eine Übersicht der 15 Elemente, aus denen Dublin Core [Dublin Core 2000] besteht:

- **Titel (DC.title)**: Titel der Quelle; der vom Verfasser, Urheber oder Verleger vergebene Name der Ressource.
- **Verfasser oder Urheber (DC.creator)**: Die Person/en oder Organisation/en, die für den Inhalt verantwortlich sind. Dies können z.B. Autoren bei Textdokumenten sein oder Künstler bzw. Maler bei graphischen Dokumenten.
- **Thema und Stichwörter (DC.subject)**: Dieses Element soll der Entwicklung eines kontrollierten Vokabulars dienen. Es enthält eine Liste von Schlüsselwörtern, die in keiner bestimmten Reihenfolge angeordnet sein müssen.
- **Inhaltliche Beschreibung (DC.description)**: Eine Beschreibung des Ressourceninhalts inklusive einer kurzen Zusammenfassung (Abstract) bei textbasierten Ressourcen oder Inhaltsbeschreibungen bei graphischen Ressourcen. Künftige Metadaten-Sammlungen können auch numerische Inhaltsbeschreibungen (z.B. Spektralanalyse einer graphischen Ressource) enthalten. Kann diese nicht in bestehende Netzsysteme eingebettet werden, kann dieses Feld einen Link zu einer solchen Beschreibung beinhalten statt der Beschreibung selbst.
- **Verleger bzw Herausgeber (DC.publisher)**: Die Einrichtung, die verantwortet, daß diese Ressource in dieser Form zur Verfügung steht und damit auch den Zugang zu dieser Ressource gewährt; wie z.B. ein Verleger, ein Herausgeber, eine Universität.
- **Weitere beteiligte Personen und Körperschaften (DC.constributors)**: Person/en oder Organisation/en, die zusätzlich zu den in DC.Creator genannten, einen bedeutsamen intellektuellen Beitrag zur Ressource geleistet haben, deren Beitrag aber eher sekundär zu denen in DC.Creator ist (z.B. Übersetzer, Illustrator, Konferenzleiter).
- **Datum (DC.date)**: Das Datum, an dem die Ressource in der gegenwärtigen Form zugänglich gemacht wurde. Der empfohlene Eintrag des Datums

ist eine achtstellige Zahl (JJJJMMTT), wie es in ANSI X3.30-1985 definiert ist. Der 29. Februar 2000 würde dann z.B. als 20000229 geschrieben werden. Andere Darstellungsschemata sind zwar erlaubt, sollten aber eindeutig identifiziert werden, damit es nicht zu Fehlinterpretationen kommen kann. Beim 4. Dublin Core Workshop entschied man sich für die Unterteilung dieses Elements in DC.date.Creation (Datum der erstmaligen Veröffentlichung im Netz) und DC.date.Lastmodified (Datum der letzten Änderung).

- **Ressourcenart (DC.type)**: Die Art der Ressource, wie z.B. Homepage, Arbeitsbericht, technischer Bericht, Roman, Gedicht, Wörterbuch, wird hier eingetragen. Es wird erwartet, daß die Angabe für dieses Feld aus einer definierten Liste von Ressourcenarten entnommen wird. Eine solche Liste findet man unter der URL http://www.roads.lut.ac.uk/Metadata/DC- ObjectTypes.html.

- **Format (DC.format)**: Hier wird das datentechnische Format der Ressource eingetragen, z.B. Text/HTML, ASCII, Postscript, JPEG, ausführbare Datei. Mit diesen Informationen ist es Mensch und Maschine ermöglicht, über die Verarbeitungsmöglichkeiten der kodierten Daten zu entscheiden. Auch hier soll die Angabe aus einer vordefinierten Liste, wie die der Internet Media Types (MIME-Types) entnommen werden. Es können grundsätzlich auch andere nicht-elektronische Medien, wie Bücher und Zeitschriften mit eingeschlossen werden.

- **Ressourcen-Identifikation (DC.identifier)**: In dieses Feld wird eine Zeichenkette oder Zahl eingetragen, die diese Ressource eindeutig identifiziert. Das können vernetzte Ressourcen wie URLs oder URNs sein oder weltweit eindeutige Identifikationen wie die ISBN (International Standard Book Number).

- **Quelle (DC.source)**: Dies Element enthält die Datenquelle (gedruckt oder in elektronischer Form), von der das Dokument abgeleitet wurde.

- **Sprache (DC.language)**: Die Sprache, in der die Ressource verfaßt ist, wird hier eingetragen. Es sollte der dreistellige Z39.53- Sprachcode verwendet werden (http://www.sil.org/sgml/nisoLang3-1994.html).

- **Beziehung zu anderen Ressourcen (DC.relation)**: Hiermit ist es möglich, Verbindungen unter verschiedenen Ressourcen darzustellen, die einen formalen Bezug zu anderen Ressourcen haben, aber als eigenständige Ressourcen existieren; z.B. Bilder in einem Dokument oder einzelne Kapitel eines Buches.

- **Räumliche und zeitliche Maßangaben (DC.coverage)**: Hier werden Angaben zur räumlichen Bestimmung (z.B. geographische Koordinaten) und zeitlichen Gültigkeit eingetragen, die die Ressource charakterisieren.

- **Rechtliche Bedingungen (DC.rights)**: Die Angaben in diesem Feld ermöglichen es Informationsanbietern, die Verbindung von Bezugs- und Benutzungsbedingungen sowie rechtlichen und abrechnungsbedingten Voraussetzungen oder Urhebervermerken mit einer entsprechenden Ressource oder einer Sammlung von Ressourcen herzustellen.

5.1.2 SGML

Textdokumente setzen sich aus den drei Bestandteilen Inhalt, Struktur und Format zusammen. Schlagen wir eine Zeitschrift oder ein Magazin auf, so ist das Dokument auf Inhalt und Format reduziert worden. Zwar erkennen wir als Leser noch die Struktur (z. B. Überschriften oder Zitate), eine maschinelle Verarbeitung des Textes ist aber praktisch nicht mehr möglich.
Gängige Textverarbeitungen oder DTP-Programme verhalten sich ähnlich. Erst wenn die Struktur extern gespeichert wird, läßt sich ein Dokument maschinell sinnvoll verarbeiten. Liegt die Struktur getrennt vor, können z. B. die Daten aus einer Rechnung direkt in ein Buchungssystem oder eine Datenbank übertragen werden. Hier setzt die Standard Generalized Markup Language (SGML) an, deren Ursprünge in den 70er Jahren liegen.

Zur Geschichte von SGML

Die wesentliche Idee von SGML, das Konzept des „generic coding“, kam im September 1967 von William Tunnicliffe von der GCA (Graphic Communications Association). Er machte den Vorschlag, den Informationsgehalt eines Dokuments von seiner äußeren Form zu trennen.
Ebenfalls Ende der sechziger Jahre veröffentlichte Stanley Rice (ein New Yorker Buch-Designer) seine Idee der „editorial structure tags“, woraus später das „generic markup“ entstand. Der Begriff „markup“ stammt aus dem Verlagswesen, aus einer Zeit, als Begriffe wie Desktop-Publishing noch unbekannt waren.
Nach der inhaltlichen Überprüfung und Korrektur eines Manuskripts folgte die Bearbeitung durch den Layouter, der die Entscheidungen über Seitenformat, Zeichensätze und weitere typographische Festlegungen traf. Diese wurden in Form von handschriftlichen Markierungen und Anweisungen in das Manuskript eingefügt und anschließend im Satz berücksichtigt.

Eine genaue Umsetzung dieser Vorgehensweise spiegelte sich dann auch in elektronischen Dokumenten wider. Texte enthielten Steuerzeichen oder Makros, die Anweisungen für die Formatierung gaben. Auch heute arbeiten die gängigen Textverarbeitungen nach diesem Prinzip. Zwar stellen WYSIWYG-Programme die Steuerzeichen nicht mehr auf dem Bildschirm dar, sondern können dank graphischer Oberflächen Attribute wie Fettdruck und verschiedene Zeichensätze unmittelbar anzeigen, die Fixierung auf das Layout des Dokuments herrscht jedoch immer noch vor.

Das Prinzip

Beim Generic Coding geht es darum, die Struktur und die logischen Elemente eines Dokumentes zu kennzeichnen. Diese Art der Information geht bei der üblichen Vorgehensweise verloren. Setzt der Verfasser eine Überschrift in einer größeren Schriftart, so kann ein Leser diesen Text zwar als Überschrift erkennen, der Computer „weiß“ aber nichts davon. Das Dokument enthält in diesem Fall nur die Information, wie ein Text darzustellen ist, aber nicht, welche Funktion er erfüllen soll.
Hier setzt das Generic Coding an. Markierungen sagen etwas über die Art (generic: artmäßig) der markierten Stelle aus. Die Art bezeichnet hier z.B. ein Kapitel, eine Überschrift, einen hervorgehobenen Text, eine Fußnote, einen Eigennamen, ein Zitat usw. Der Vorteil liegt also darin, daß die Struktur des Dokuments bei der Speicherung nicht verloren geht.
Die Zuordnung einer bestimmten Darstellung zu einer bestimmten Klasse von Textstellen ist anwendungsspezifisch eindeutig möglich: bei einem Ausdruck etwa hätten alle Überschriften die gleiche Schriftart und -größe. Das waren die ursprünglichen Ideen von Tunnicliffe und Rice. Die Bedeutung dieser Idee erkannte Norman Scharpf, Direktor der GCA. Er gründete ein Kommitee, dessen Entwicklung des „GenCode“-Konzepts später Bedeutung für SGML erlangte.
Ebenfalls auf den Ideen von Tunnicliffe und Rice basierend, entwickelten Charles Goldfarb, Edward Mosher und Raymond Lorie bei IBM 1969 die Generalized Markup Language (GML), die erstmals das Konzept eines formal definierten Dokumenttyps mit einer verschachtelten Struktur enthielt. Die Entwicklung eines Standards begann 1978, als das ANSI (American National Standard Institute) ein Kommitee gründete, dessen Ziel die Entwicklung einer standardisierten Textbeschreibungssprache auf GML-Basis war. Goldfarb war Mitglied dieser Gruppe, die auch vom GCA-GenCode-Kommitee Unterstützung erhielt. Nach mehreren Normentwürfen fand 1984 eine Neuorganisation des Projekts unter Leitung des ANSI und auch der ISO (International Organisation for Standardization) statt.

Ein letzter Entwurf für den internationalen Standard wurde im Oktober 1985 präsentiert. Nach einem Jahr, in dem man Anregungen und Kommentare einarbeitete, gipfelte die Arbeit 1986 in der Veröffentlichung der SGML als ISO-Standard 8879.
Nimmt man sich die einzelnen Bestandteile des Begriffs SGML und analysiert sie, so läßt sich schon einiges über diese Markup-Sprache erkennen:

- **Standard**: SGML ist durch verschiedene Gremien standardisiert worden (1986: ISO 8879, 1990: EN 28879, 1991: DIN 28873), ist erweiterbar, plattform- und geräteunabhängig.
- **Generalized**: SGML ist beschreibend; mit ihrer Hilfe sollen allgemeingültige Strukturen abgebildet werden.
- **Markup**: SGML ist eine Auszeichnungssprache.
- **Language**: SGML ist eine formal definierte Metasprache, die Dokumentformate beschreibt.

Grundkonzepte von SGML:

- **Dokumenttypen**: Dazu gehören die Dokumenttypdefinition (DTD) und die Dokumenttypdeklaration am Anfang eines Dokuments.
- **Elemente**: Ein Element wird in SGML durch Start- und Endtags (tag = Marke) eingeschlossen. Formal sieht das so aus:
  <*elementName*> elementInhalt </*elementName*>.
  Die Elemente müssen in einer DTD definiert werden. Elemente können sich nicht überlappen.
- **Inhaltsmodelle**: Strukturelle Elementinhalte werden über sogenannte Inhaltsmodelle abgebildet, die die impliziten Konventionen der realen Welt repräsentieren sollen. Ein Beispiel hierfür ist, daß implizit angenommen wird, daß man bei einem Element Empfänger die Angaben des Namens, gefolgt von der Strasse oder eines Postfachs und des Ortes erwartet. Inhaltsmodelle sind syntaktisch gesehen reguläre Ausdrücke mit folgenden Operatoren:

  ```
  Konnektoren: Sequenz(,), Und(&), Oder(|)
  Häufigkeitsindikatoren: beliebig oft(*), mindestens einmal(+), optional(?)
  ```

  So kann eine Empfängerangabe folgendermaßen aussehen:

```
Firmenname, Ansprechpartner?, (Strasse | Postfach)?, (PLZ,Ort)
```

- **Attribute**: Man kann jedem Element einen Attributnamen und einen Wertetyp zuweisen. Dies kann z.B. sinnvoll sein, wenn man in seinem Text verschiedene Sprachen verwendet und die Rechtschreibprüfung auf diesen Umstand hinweisen möchte.

- **Einheiten/Entitäten**: Für folgende Probleme wurde ein einheitliches Entitätskonzept entwickelt:

  - *Zerlegung eines Projektes in Teilprojekte.* Gerade in einem größeren Projekt ist es erforderlich, ein grosses Dokument in mehrere kleine Dokumente aufzuteilen, da sie von verschiedenen Personen bearbeitet werden. Es soll aber gleichzeitig ein Gesamtdokument erhalten bleiben.
  - *Textbausteine.* In einigen Fällen kommen immer wiederkehrende Phrasen oder Dokumentteile vor, die aus Gründen der einfachen Änderbarkeit an einer zentralen Stelle verwaltet werden sollen.
  - *Systemabhängige Sonderzeichen.* Möchte man Dokumente zwischen verschiedenen Systemen austauschen und in diesen sind Sonderzeichen enthalten, muß man das Problem der unterschiedlichen Zeichensätze lösen. Eine einfache Möglichkeit ist die Vergabe von symbolischen Namen in einem vereinbarten Grundzeichensatz (z.B. ISO 646). Die symbolischen Namen werden dann auf dem fremden System richtig umgesetzt.
  - *Integration von Binärdaten.* Hierbei stellt sich die Frage, wie das System angewiesen werden kann, an einer bestimmten Stelle im Dokument eine Binärdatei einzufügen.

  Das Grundkonzept besteht darin, einem symbolischen Namen eine Dokumenteinheit zuzuweisen (z.B. einen Ersetzungstext) und dann im Dokument selbst nur noch den symbolischen Namen zu verwenden.
  Nach der Art der Dokumenteinheit kann man die Verwendung des Entitätskonzepts klassifizieren:

  - *externe Entitäten.* Man gibt als Dateneinheit, die für den symbolischen Namen eingesetzt werden soll, eine Datei an, wie z.B. Standarddefinitionen, Binärdaten, Subdokumente.
  - *Textentitäten.* Die Dateneinheit ist hier ein Ersetzungstext, der auch als systemabhängig markiert werden kann (z.B. Sonderzeichen).

  - *Parameterentitäten.* Sie können nur innerhalb von DTD (Dokumenttypdefinitionen) verwendet werden und stellen eine einfache Abkürzung dar. Dies wird hauptsächlich dazu benutzt, vordefinierte Entitätssätze einzubinden, die dann eine ganze Liste von Sonderzeichen enthalten.

- **öffentliche Bezeichner**: Möchte man systemübergreifend auf Standarddokumente verweisen, verwendet man öffentliche Bezeichner, die eine Namenskonvention für Dokumentteile darstellen. Wesentliche Elemente sind:
  - die verantwortliche Organisation
  - der Typ des Dokumentteils (DTD, ENTITIES usw.)
  - der standardisierte Name
  - die Sprache

- **Notationen**: In SGML gibt es die Möglichkeit, Schreibweisen zu definieren, die für jede Notation eine verarbeitende Anwendung angeben. So kann man z.B. die TeX-Schreibweise von Formeln in ein SGML-Dokument einfügen, das sonst über keine Möglichkeit der Darstellung von Formeln verfügt.

- **Konkrete Syntax**: Intern arbeitet SGML mit einer abstrakten Syntax, die in der sog. SGML-Deklaration auf eine konkrete Syntax (z.B. ISO 7-Bit und 8-Bit, EBCDIC, Unicode) abgebildet werden kann.

5.1.3 HTML

Eine der bekanntesten Anwendungen von SGML ist die Hypertext Markup Language (HTML), die Sprache des World Wide Web.
HTML ist eine einfache Auszeichnungssprache, die aus nicht ganz sechzig Elementtypen (Tags) wie `<b>` oder `<table>` besteht. Durch die HTML-Elemente läßt sich die Basisstruktur von Texten (Überschriften, Absätze, Zitate, Listen, Tabellen usw.) abbilden.
Der ursprüngliche Sprachumfang von HTML wurde seinerzeit von Tim Berners-Lee am CERN (Hochenergie-Physik-Institut in der Schweiz) festgeschrieben.
Durch die rasante Entwicklung des WWW forderten aber Industrie und Webdesigner bald Erweiterungen, die Mitte der 90er zu einem richtigen Tag-Krieg der beiden großen Browserhersteller Netscape und Microsoft ausartete und zueinander inkompatible Elemente wie `BLINK` (ein nur von Netscape-Browsern verstandenes Element zur Erzeugung eines blinkenden Textes) oder `MARQUEE` (proprietäres Microsoft-Tag für eine Laufschrift) hervorbrachten.

Neben diesen proprietären Elementen wurde HTML auch immer mehr durch Tags mit Formatierfunktionen verwässert. Kaum eine Webseite nutzt heute strukturbeschreibende Elemente wie `<strong>`, `<cite>` oder `<sample>`. Vielmehr kommen hier die Formatiertags `<b>`, `<i>` oder `<tt>` zum Einsatz.
Mit der Einführung der Cascading Style Sheets (CSS) versuchte das World Wide Web Consortium (W3C) eine Rückbesinnung auf die ursprünglichen Ziele von HTML, die Trennung von Struktur und Form. Mit CSS läßt sich die Form, also das Aussehen eines Elementes definieren. Eine CSS-Anweisung wie

```
h1 {
  font-family:Arial;
  color:blue;
  font-weight:bold;
  left-margin:15mm;
}
```

weist einer Überschrift der ersten Stufe ein blaue, fette und serifenlose Schrift zu, die um 1,5 cm vom linken Seitenrand einzurücken ist.
CSS ist hier so fortschrittlich, daß die Formatieranweisungen dem Ausgabemedium angepaßt werden können, je nach dem, ob die Ausgabe nun in einem Browser, auf einem Drucker oder akustisch (sog. aural style sheets) erfolgt.
Dublin Core Elemente können sehr gut in HTML Dokumente integriert werden. Die Einbettung erfolgt zwischen den <HEAD>-Tags. Das folgende Beispiel soll einen kurzen Eindruck davon vermitteln:

```
(nach: http://purl.org/metadata/dublin_core The Dublin Core
Metadata Element Set).
<HTML>
<HEAD>
<TITLE>UKOLN: UK Office for Library and Information Networking</TITLE>
<META NAME="DC.title" CONTENT="UKOLN: UK Office for Library and Information
      Networking">
<META NAME="DC.subject" CONTENT="national centre, network information
support,
      library community, awareness, research, information services, public
      library networking, bibliographic management, distributed library
      systems, metadata, resource discovery, conferences, lectures,
      workshops">
<META NAME="DC.description" CONTENT="UKOLN is a national centre for support
in
      network information management in the library and information
      communities. It provides awareness, research and information
      services">
<META NAME="DC.creator" CONTENT="UKOLN Information Services Group">
```

```
</HEAD>
<BODY>
...
</BODY>
</HTML>
```

Für die Arbeit mit Metadaten unter HTML sind unterschiedliche Ansätze denkbar:

- Die Metadaten werden beim Erzeugen des Dokuments eingefügt. Dies geschieht per Hand in einem Editor. Diese sehr einfache Lösung hat folgende Nachteile:
  - Der Trend bei HTML-Editoren geht immer mehr zu WYSIWYG. Deshalb sollte man aus Konsistenzgründen vermeiden, vom Benutzer zu verlangen, teilweise im Quellcode und teilweise in einer graphischen Umgebung arbeiten zu müssen. Zudem kann es sein, daß der Benutzer bei bestimmten Werkzeugen den Quelltext gar nicht mehr zu Gesicht bekommen kann.
  - Die Instandhaltung von großen Datenmengen mit integrierten Metadaten kann problematisch werden, wenn sich etwas an der Syntax ändert.
  - Wenn sich der Benutzer vertippt oder die Elemente in einer inkorrekten Art und Weise benutzt, kann es zu falschen Interpretationen der Metadaten kommen.
- Die Metadaten werden beim Veröffentlichen des Dokuments eingefügt. Hierbei finden Management Tools zur Erstellung von Websites Verwendung. Typischerweise halten sie alle Informationen, die für die Site von Bedeutung sind, in einer Datenbank bereit. Bei Bedarf werden die HTML-Seiten dann automatisch erstellt. Ein Nachteil dieses Ansatzes besteht darin, daß man sich hier auf ein bestimmtes System festlegt und keine anderen Formate mehr verwendet werden können.
- Die Metadaten werden beim Aufruf der Seite eingefügt. Es werden Server Side Include (SSI)-Skripte verwendet, die die Metadaten, die separat in einem neutralen Format gespeichert vorliegen, bei Bedarf einbetten. Der Nachteil liegt hier in der Verarbeitungsgeschwindigkeit, denn jede Seite, die aufgerufen wird, muß dahingehend überprüft werden, ob zu dem HTML-Dokument ein SSI-Skript existiert und dann bei Bedarf ausgeführt werden.

5.1.4 XML

Mit der rasanten Entwicklung des World Wide Web und HTML als Auszeichnungssprache wurden auch schnell die Unzulänglichkeiten von HTML an den Tag gebracht:

- fest definierter Sprachumfang
- Vermischung von Markup und Layout (z.B. `<EM>` vs. `<I>`)
- proprietäre Erweiterungen seitens der beiden großen Browserhersteller Microsoft und Netscape (z.B. `<BLINK>` oder `<MARQUEE>`)

Gerade der erste Punkt schränkt das Einsatzgebiet von HTML für sinnvolle Anwendungen stark ein und nicht von ungefähr liefern Suchmaschinen wie Alta Vista oder Fireball bei Recherchen kaum brauchbare Ergebnisse aus der Informationsflut (oder besser aus der Flut der Nicht-Informationen). So kann eine Suchmaschine nicht unterscheiden, ob auf einer Webseite ein Stellenangebot ausgeschrieben ist oder ob es sich um einen aktuellen Nachrichtenartikel handelt. Mit XML können diese Informationen mitgegeben und von einer Suchmaschine ausgewertet werden.
Zum letzten Punkt gibt es ein bekanntes Zitat von Jim Cape aus der Newsgroup comp.infosystems.www.authoring.html (zitiert aus [Behme et al. 1998, S. 4] nach [Connoly et al. 1997]):

> „XML wurde entwickelt, um ein für alle mal die von Microsoft und Netscape propagierten Tag-Suppenkriege zu beenden."

Die Frage, was man nicht alles mit Web-Dokumenten anstellen könnte, wenn die Grenzen von HTML (=feststehender Satz von Elementen) nicht für alle Dokumente gelten würden (weil es eben keine eigendefinierten Elemente gibt), ließen weder das W3C noch die SGML-Anhänger ruhen. So wurde Ende 1996 im Rahmen einer SGML-Veranstaltung der erste Vorschlag zu XML vorgestellt. Schon im April 1997 war XML eines der meistdiskutierten Themen der WWW-Konferenz 1997.
Bei der Entwicklung von XML standen die folgenden Punkte im Vordergrund (vgl. [Behme et al. 1998]):

- XML soll sich im Internet auf einfache Weise nutzen lassen.
- XML soll ein breites Spektrum von Anwendungen unterstützen.
- XML soll zu SGML kompatibel sein.

- Es soll einfach sein, Programme zu schreiben, die XML-Dokumente verarbeiten.
- Die Anzahl optionaler Merkmale in XML soll minimal sein, idealerweise Null.
- XML-Dokumente sollten für Menschen lesbar und angemessen verständlich sein.
- Der XML-Entwurf sollte zügig abgefaßt werden.
- Der Entwurf von XML soll formal und präzise sein.
- XML-Dokumente sollen leicht zu erstellen sein.
- Knappheit von XML-Markup ist von minimaler Bedeutung.

Der letzte Punkt ist deshalb nicht erstrebenswert, weil Maschinen nun einmal Mengendaten gut verarbeiten können und Dokumente schon jetzt häufig mit Hilfe von Datenbanken automatisch (database publishing) erstellt werden.
Damit Programme (Parser) XML-Dokumente einfach verarbeiten können, muß bei der Erstellung von XML-Dokumenten stärker auf die Wohlgeformtheit geachtet werden. Hierzu zählt, daß

- Elemente immer geschlossen werden (d.h. zu jedem Starttag muß es auch ein Endtag geben)
- Attributwerte immer in Anführungszeichen zu setzen sind
- Elemente nur einfach geschachtelt werden dürfen.

Die letzten beiden Punkte werden auch von HTML gefordert, jedoch sehen viele Browser über diese Fehler hinweg – natürlich mit unterschiedlichen Ergebnissen im Layout und Inkompatibilitäten zu Folgeversionen.
XML steht für *eXtensible Markup Language* und um eine Erweiterung kommt man nicht herum, da XML keinen Grundwortschatz bietet. Aus diesem Grund wird zu Beginn eines XML-Dokumentes auch dessen DTD (document type definition) hinterlegt, entweder direkt in das Dokument eingebettet oder als URL.
Eine DTD beschreibt den strukturellen Aufbau und die logischen Elemente einer Klasse von Dokumenten, genannt Dokumenttyp.
Beispiel: Eine E-Mail bestehe aus: Empfänger, Absender, Thema, Nachricht. Eine DTD könnte wie folgt aussehen:

```
<!-- E-Mail-DTD -->
<!ELEMENT email         (empfaenger, absender, thema, nachricht)>
<!ELEMENT empfaenger    (#PCDATA)>
<!ELEMENT absender      (#PCDATA)>
<!ELEMENT thema         (#PCDATA)>
<!ELEMENT nachricht     (#PCDATA)>
```

Alle Elemente enthalten reinen Text, Parsed Character Data (PCDATA) genannt. Das eigentliche Dokument könnte so aussehen:

```
<?xml version="1.0"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
        <empfaenger>Andrea</empfaenger>
        <absender>Birgit</absender>
        <thema>Wie geht's?</thema>
        <nachricht> Hallo,
                    ich wollte mich nur einmal kurz bei Dir melden.
                    Viele Gruesse, Birgit.
        </nachricht>
</email>
```

Es ist in XML erlaubt, eine Instanz ohne DTD zu schreiben. Das ist von Nutzen, wenn man nur eine einzige Instanz schreiben möchte. Hat man aber mehrere gleichartige Texte (z.B. eine Reihe von E-Mails), dient eine DTD als Muster, als Vorlage. Ein Editor kann dieses Muster schon beim Schreiben mit dem aktuellen Dokument vergleichen und auf strukturelle Fehler hinweisen (z.B. ein fehlender Empfänger). Benutzt man keine DTD kann man von keinem Programm erwarten, dass es „weiß“, welche Bestandteile eine Instanz besitzen muß.
Ein wesentlicher Vorteil des XML-Ansatzes besteht darin, daß sich einmal geschriebene Dokumente in vielfältiger Weise verwenden lassen: Je nach Bedarf kann man sie in HTML für das Web-Publishing, in RTF für die Benutzung mit einer Textverarbeitung oder in PostScript für die Druckausgabe konvertieren. Ebenso für die Einspeisung in eine Datenbank. Diese Flexibilität spart Kosten und erhöht die Lebensdauer der Daten.
Die Dokumenttyp-Deklaration informiert den XML-Prozessor darüber, welche Dokumenttyp-Definition (DTD) für die folgende Instanz gültig ist und entweder wo sich die Datei mit der DTD befindet (externe DTD) oder wie die DTD aussieht (interne DTD). Die DTD ist damit ein Bestandteil der Dokumenttyp-Deklaration. Informell beschrieben enthält eine DTD die Informationen darüber, welche Elementtypen es gibt, welchen Inhalt sie haben dürfen, welche Attribute erlaubt sind und welche Werte sie annehmen dürfen.

Die Tatsache, daß es keinen XML-Grundwortschatz gibt, macht zudem eine strikte Trennung von Struktur und Form des Dokuments erforderlich. Kein XML-Browser der Welt kann wissen, wie er obiges Dokument darstellen soll (außer vielleicht als hierarchische Liste). Die Informationen über das „Wie“ wird durch Style Sheets definiert. Da XML eine echte Teilmenge von SGML ist, kann hier DSSSL (Document Style Semantics and Specification Language) eingesetzt werden.
Im wesentlichen besteht DSSSL aus zwei Komponenten: der Transformationssprache (transformation language) und der Stilsprache (style language). Erstere soll (SGML-)Dokumente, die nach einer bestimmten DTD (oder mehreren) kodiert sind, in ein Format wandeln, in dem sie anderen DTDs entsprechen. Die zweite (Stil) legt für Teile eines Dokuments fest, wie sie aussehen sollen. Wichtige Bestandteile von DSSSL sind die Flow Object Classes, die Standard Document Query Language (SDQL) und die Expression Language, die sich insofern überlappen, als etwa die Expression Language sowohl in SDQL als auch in der Transformationssprache und der Stilsprache vorkommt.
Ein DSSSL-Programm legt im wesentlichen zwei Dinge fest:

- Was soll verarbeitet werden?
- Wie soll es ausgegeben werden?

Dies ist die aus CSS bekannte Vorgehensweise: Einem Element wird ein Aussehen zugeordnet. In DSSSL nennt man das eine Konstruktionsregel. Stark vereinfacht kann man eine solche Regel wie folgt schreiben: `(was (wie))`. Die Schreibweise mit den Klammern stammt daher, dass DSSSL auf Scheme, einem Lisp-Dialekt, basiert (was auch Programmierung möglich macht).
Das Ergebnis der Umwandlung eines XML-Dokumentes unter Anwendung der DSSSL kann ein HTML-Dokument, aber auch ein RTF- oder ein Postscript-Dokument sein. Auch die Ausgabe auf andere Medien (z. B. Audioausgabe) ist möglich.
Der Einsatz von DSSSL ist jedoch für viele gestandene Web-Autoren zu kompliziert, daher soll XSL (eXtensible Sytle Language) die Stilsprache zu XML werden. XSL läßt sich auch durch den Einsatz von ECMA-Script (die standardisierte Version von JavaScript) programmieren. Die folgenden Designziele wurden bei XSL verfolgt (siehe [Behme et al. 1998]):

- XSL sollte sich über das Internet leicht nutzen lassen.
- XSL sollte in XML-Syntax ausgedrückt sein.

- XSL sollte eine deklarative Sprache bieten, um normale Formatierungen vorzunehmen.
- XSL sollte einen „Ausweg“ in eine Skript-Sprache bieten, für komplexere Formatierungen und um Erweiterbarkeit und Vollständigkeit möglich zu machen.
- XSL wird eine Untermenge von DSSSL mit der vorgeschlagenen Ergänzung sein.
- Die mechanische Abbildung eines CSS- in ein XSL-Style-Sheet sollte möglich sein.

XSL ist derzeit darauf ausgerichtet, XML-Daten in HTML zu wandeln. Jedes XSL-Style-Sheet besteht – ähnlich den DSSSL-Spezifikationen – aus Vorgaben, wie Elemente darzustellen sind. Diese Vorgaben erfolgen durch Konstruktionsregeln und Stilregeln.

XML als Metadatum

Im Vergleich zu HTML bietet XML mehr Möglichkeiten, da es eine Metasprache ist. Die Element- und Attributnamen bzw. deren Werte sind die Metadaten und enthalten Informationen über den eigentlichen Inhalt.
Eine Alternative zu XML wäre auch SGML gewesen, jedoch ist SGML für die Bedürfnisse des Web zu kompliziert. Da sich mit XML eigene Sprachen definieren lassen, ist XML quasi das „neue“ SGML.
XML bietet sich darüber hinaus auch als universelles Datenformat an, das sowohl einfach maschinell verarbeitbar als auch für menschliche Augen verständlich ist. Beispiel: Laut Signaturgesetz müsste z.B. einem HBCI[1]-Anwender die zu signierende Nachricht an das Kreditinstitut präsentiert werden. Aber würde ein Anwender eine Ausgabe wie

```
HKUEB:3:2+12345678+70230000+ERWIN MUSTERMANN+EUR+100,+51+RECHNUNG NR. 4711
```

unterschreiben? Mit XML wäre schnell verständlich, worum es hier geht:

```
<Einzelüberweisung>
  <Empfänger>
    <Name>Erwin Mustermann</Name>
    <Kontonummer>12345678</Kontonummer>
    <Bankleitzahl>70230000</Bankleitzahl>
  </Empfänger>
```

[1] HBCI steht für Homebanking Computer Interface

```
    <Gutschrift>
       <Währung>EUR</Währung>
       <Betrag>100,00</Betrag>
    </Gutschrift>
    <Verwendungszweck>Rechnung Nr. 4711</Verwendungszweck>
  </Einzelüberweisung>
```

Erweiterungen zu XML

Die Flexibilität von XML DTDs zu kombinieren birgt das Problem doppelter Element-Bezeichner. In XML wurde dieses Problem durch die Einführung von Namensräumen gelöst.
XML Schemata definieren, vereinfacht gesagt, Datentypen für die Elementinhalte. Mit einem entsprechenden Schema kann also definiert werden, wie ein Betrag zu schreiben ist und welche Werte zulässig sind (siehe [W3C 2000b, W3C 2000c]).

5.1.5 RDF

RDF (Resource Description Framework) ist die aktuellste Anstrengung des W3C zur Bewältigung des Metadaten-Problems. So soll ihre Spezifikation zukünftig als Grundlage zur Benutzung von Metadaten in internet-basierten Informationssystemen dienen [Lassila et al. 1999].
Auch Tim Berners-Lee hat im Mai 1999 auf der achten WWW-Konferenz in seiner Rede *Challenges of the Secon Decade* darauf aufmerksam gemacht, dass zur verbesserten Informationsverarbeitung ein semantisches Web notwendig ist. Die Grundlage hierfür sieht er in RDF [Berners-Lee 1999].
Das W3C sieht als Einsatzgebiete für RDF eine Reihe von Anwendungen:

- die Verbesserung von Ergebnissen aus Suchmaschinen
- Katalogisierung und Beschreibung der Zusammenhänge von Daten einer Webseite
- digitale Bibliotheken
- intelligente Agenten, die Wissen austauschen
- Bewertung von Inhalten einer Webseite durch Dritte (Content Rating)
- Beschreibung einer Menge von Web-Ressourcen, die zusamengenommen ein logisches Dokument darstellen

- Annotationen von Copyright-Informationen
- Definition von Benutzer-Präferenzen und Präferenzregeln einer Webseite, die personalisierten Informationsaustausch erlauben.

Mit RDF können PICS-Label (Platform for Internet Content Selection) ausgedrückt werden, was auch in der Absicht der RDF-Autoren lag, die sich von den Erfahrungen mit PICS inspirieren ließen. RDF soll PICS zukünftig ersetzen. PICS ist eine Empfehlung des W3C zur Definition von Filterregeln. Sie verhindern durch das Auswerten von Metadaten, daß z.B. bestimmte Web- Dokumente Kindern zugänglich gemacht werden.
Der auf PICS aufbauende DSig-Standard erlaubt es, Dokumente digital zu unterschreiben. Laut dem W3C bildet RDF zusammen mit digitalen Unterschriften die Grundlage für ein „Web of Trust“, das für den Electronic Commerce und kollaborative Applikationen nötig ist.
Das Datenmodell von RDF entspricht einem gerichteten Graphen, der aus Knoten und Pfeilen besteht. Einem Knoten entspricht entweder eine Web-Ressource oder ein Literal (ähnlich einer Zeichenkette). Jedes Internet-Objekt, das eindeutig durch einen URI (uniform resource identifier) identifiziert werden kann, ist eine solche Ressource (eine HTML-Seite, ein Teil einer solchen Seite, ein Bild, eine Audio- oder Videodatei). Ein Pfeil innerhalb des Graphen repäsentiert ein Property, mit dem eine Eigenschaft bzgl. einer Informationsquelle definiert wird. Mittels XML-Namespaces werden RDF-Vokabulare referenziert, die Properties für bestimmte Anwendungsgebiete definieren. RDF-Properties können ausdrücken, in welcher wechselseitigen Beziehung die Ressourcen stehen. Insofern ähnelt ein RDF-Datenmodell einem Entity-Relationship Diagramm.
Es folgt ein einfaches Beispiel, das ein Gefühl dafür vermitteln soll, wie RDF arbeitet, und um zu zeigen, wie ähnlich sich die Konzepte der RDF-Schemata (schemas) und XML-Namensräume (namespaces) sind:

```
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns# "
    xmlns:DC="http://purl.org/dc/elements/1.0/ ">
<Description about="http://www.labusch.de ">
<DC:title>Homepagge von Birgit Labusch </DC:title>
<DC:creator>Birgit Labusch </DC:creator>
<DC:date>1999-10-20 </DC:date>
<DC:subject>Studienarbeit, JavaScript, Windowsoberfl"achengestaltung,
    Kreditwürdigkeitsprüfung mit Bilanzen, Umweltinformatik, VRML
</DC:subject>
</Description>
</RDF>
```

Das Grundkonzept besteht darin, daß Metadaten über Webinhalte durch eine Sammlung von Eigenschaften, die sich RDF Description nennen, beschrieben werden. Dabei benutzt RDF die Syntax von XML; wobei zu erwähnen ist, daß RDF auch in XML beschrieben wurde.
Zeile 1 des obigen Beispiels deklariert das ganze XML-Element als einen RDF-Ausdruck, dessen Format durch das RDF-Modell und die Syntax- Spezifikation definiert wird.
Zeile 2 gibt an, wo das Vokabular für RDF im Netz zu finden ist. Hier wird Dublin Core verwendet. Der Namensraum für Dublin Core erhält das Präfix DC und kann in den folgenden Elementen verwendet werden.
In Zeile 3 wird angegeben, daß sich diese Metadaten auf das Dokument beziehen, das durch das `about`-Attribut angegeben wird.
In Zeile 4 bis 7 stehen die eigentlichen RDF-Ausdrücke, die die Metadaten ausmachen. Sie entstammen dem Dublin Core RDF-Vokabular. Wenn die Metadaten bearbeitet werden, erkennt die Software diese Auszeichner und kann entsprechend mit ihnen verfahren.
Mit Hilfe von RDF-Schemata kann das Vokabular der Metadatenbeschreibung an eigene Bedürfnisse (z.B. der von Organisationen) angepaßt werden. Ein Schema definiert die Bedeutungen, Charakteristika und Beziehungen einer Menge von Eigenschaften und ggf. eine Einschränkung der möglichen Werte oder auch die Vererbung von Eigenschaften aus anderen Schemata. Die RDF- Sprache erlaubt es jedem Dokument über Metadaten die Beziehung zu dem verwendeten Vokabular durch eine Webadresse herzustellen. Die Sprachspezifikation für Schemata wurde u.a. durch die Ideen der Wissensrepräsentation (semantische Netze, Prädikatenlogik usw.) und Datenbankschemata beeinflußt. Eines der bekanntesten Schemata ist das Dublin Core.
Praktischer Nutzen von RDF [W3C 2000a]:

- **Stichwortkataloge und Bibliothekssignaturen**
  Dies sind bekannte Beispiele eines hierarchischen Systems zur Repräsentierung von Subjekttaxonomien (subject taxonomies) im Sinne von Beziehungen zwischen benannten Konzepten. Die RDF Schema-Spezifikation (RDF Schema specification) hat genau die Fähigkeiten zur Erstellung von RDF-Modellen, die die logische Struktur von Stichwortkatalogen und Bibliotheks-System-Klassifikationen repräsentieren.

- **Web sitemaps**
  Eine sitemap ist eine zusammenfassende, strukturierte Beschreibung einer Website. Die RDF Schema-Spezifikation liefert einen Mechanismus zur Definition des Vokabulars, das hierfür benötigt wird. Mit RDF läßt sich be-

schreiben, wie die Beziehungen der Seiten untereinander sind, welche Seite auf eine andere folgt usw.

- **Beschreibung der Inhalte von Webseiten**
  Dies ist eine Basisfunktionen der Dublin Core-Initiative. Dublin Core beinhaltet 15 Elemente, die in starkem Zusammenhang mit bibliographischen Informationen stehen. Sie können benutzt werden, um Ressourcen im Netz besser zu beschreiben, und es so Suchmaschinen und anderer Software zu ermöglichen, bessere Ergebnisse zu erzielen. Die Dublin Core-Workshops haben einen großen Einfluß auf die Entwicklung der RDF genommen.

- **Formale Beschreibung der Mechanismen zum Schutz persönlicher Informationen**
  Viele Webangebote bieten dem Nutzer erst nach Registrierung den vollen Funktionsumfang (z.B. Auktions-Sites, Online-Banken usw.). Hier stellen sich folgende Fragen:

  – Wie behandelt eine Website persönliche Informationen?
  – Werden irgendwelche dieser Informationen öffentlich gemacht?
  – Was erhält der Benutzer als Gegenleistung?

  Das W3C mit seinem Platform for Privacy Preferences Project arbeitet an einer Plattform, die es dem Benutzer erlaubt, zu erfahren, welche Schutzmechanismen die Site bietet. Benutzer können so ihre Bedenken ausräumen, wenn sie vorher einen Privatsphärenstatus mit dem System/der Seite ausgehandelt haben, was die Basis für eine weitere, sinnvolle Arbeit mit dem System darstellt. RDF kann u.U. dazu verwendet werden, die formale Struktur der Mechanismen zum Schutz persönlicher Informationen (privacy practice description) zu beschreiben.

- **Beschreibung der Eigenschaften von mobilen Geräten**
  Die W3C Mobile Access Activity beschäftigt sich mit einem Protokoll zur Beschreibung der Eigenschaften mobiler Geräte mit Internet-Anschluß, wie z.B. Display-Eigenschaften oder Prozessor-Leistungsfähigkeit. RDF liefert einen Weg, um die Leistungsfähigkeiten und Vorlieben der Benutzer und ihrer Hard- und Software für den Web-Zugriff zu beschreiben. Dadurch können Web-Angebote auf die spziellen Bedürfnisse der Benutzer maßgeschneidert werden (z.B. WAP (Wireless Application Protocol)).

- **Selbstkontroll- und Bewertungssystem**
  Dies bietet einen Weg, um Ressourcen einzustufen, damit Personen (oder

Rechner) ihre Informationen filtern können. RDF ermöglicht es Programmierern, sich Kontrollmechanismen für jeden erdenklichen Bereich zu schaffen (ähnlich wie mit dem PICS).

- **Metadaten mit Metadaten ausdrücken**
  Angenommen, man hat ein RDF-Bewertungssystem für Restaurants. Nun läßt sich RDF für die Metadatenbeschreibung dieses Bewertungssystems nutzen, z.B. für Datum oder Organisation. Es lassen sich also Dinge ausdrücken wie: „Der *Michelin-Restaurantführer* sagt, das Hotel Prem hat ein 5-Sterne Restaurant."

- **Digitale Signaturen**
  So wie sich das Web entwickelt und vermehrt alltägliche Aufgaben mit ihm erledigt werden, so werden auch digitale Signaturen immer wichtiger. RDF kann dazu verwendet werden, Informationen darüber auszudrücken, was man signieren möchte, welche Bedeutung die Signatur hat, den Zeitraum, in dem die Signatur gültig ist usw.

5.2 „Versteckte“ Metadaten

In diesem Teil meiner Arbeit werde ich mich mit dem Konzept der (Meta-)Datenstrukturen beschäftigen. Auf die folgenden Metadatenstrukturen werde ich näher eingehen:

- Feld (Array),
- Verbund (Record),
- Menge (Set),
- Liste,
- Keller (Stack),
- Schlange (Queue),
- Baum.

Grundlegende Strukturen, die von ihrer Art her statisch aufgebaut sind, sind das Array (Feld), der Record (Verbund), der Set (Menge) und die Sequenz (Folge). Kompliziertere Strukturen werden meist nicht als statische Typen definiert, sondern während ihrer Benutzung dynamisch generiert; dabei können sich Größe und Form verändern. Solche Strukturen umfassen Listen, Ringe, Bäume und allgemeine endliche Graphen.

Feld

Ein Feld oder Array ist eine homogene Struktur; alle seine Komponenten sind vom gleichen Typ, dem so genannten Grundtyp. Das Array ist zudem eine *random access* Struktur, was bedeutet, dass auf alle Komponenten direkt und gleich schnell zugegriffen werden kann. Um eine einzelne Komponente zu bezeichnen, wird der Name der gesamten Struktur um den *Index* zur Wahl der Komponente erweitert. Dieser Index muss ein Wert des Indextyps des Arrays sein. Die Definition eines Array-Typs T spezifiziert deshalb sowohl einen Komponententyp T0 als auch einen Indextyp T1 (die im folgenden verwendeten Notationen sind an die Programmiersprache Modula-2 angelehnt):

```
TYPE T = ARRAY T1 OF T0
z.B.: TYPE Zeile = ARRAY[1..5] OF INTEGER
```

In einem Array kann auf jedes Element gleich schnell zugegriffen werden. Dies wird über den Index, den jedes Element besitzt, ermöglicht. Der Index ist ein Datentyp mit (meist) endlich vielen Elementen. Als Index verwendet man häufig einen Unterbereich der ganzen Zahlen oder ein kartesisches Produkt von Unterbereichen. Besteht der Index nur aus einem Unterbereich, so wird das Feld als 1-dimensional bezeichnet. Ist der Index das kartesische Produkt aus N Unterbereichen, so heißt das Feld N-dimensional.
Eine wesentliche Eigenschaft von Feldern ist, dass ihre Größe feststeht und bekannt sein muss, bevor sie verwendet werden. Außerdem besitzt ein Feld eine feste Anzahl von einzelnen Daten, welche zusammenhängend gespeichert werden. Die Elemente in einem Array kennen also ihre potentiellen Vorgänger und Nachfolger, weil sie ihre Größe kennen und wissen, dass in einem Array alle Elemente den Speicher sequentiell belegen (und zwar statisch; d.h. vor der Benutzung muß für jedes Element Speicherplatz alloziiert (reserviert) werden, damit der Wert der einzelnen Elemente dort gespeichert werden kann). Eine übergeordnete Struktur hat Kenntnis über die Anzahl der Elemente und weiß, welches das erste und welches das letzte Element ist.
Mit Hilfe der Indices kann man auch Daten in Tabellenform strukturieren. So würde eine Tabelle, die eine Zeile und eine Spalte besitzt, als ein zweidimensionales Feld mit zwei Indices dargestellt werden, einem für die Zeile und einem für die Spalte. Auf diese Weise können beliebig viele Dimensionen dargestellt werden; pro Dimension kommt einfach ein weiterer Index hinzu.

Verbund

Die allgemeinste Methode zur Erstellung strukturierter Typen ist das Zusammenfassen von Elementen willkürlicher, möglicherweise selbst strukturierter Typen zu einer Einheit. In der Datenverarbeitung kommen zusammengesetzte Typen, wie Personen- oder Sachbeschreibungen, gewöhnlich in Files oder Dateien vor und registrieren (*to record*) die wichtigsten Eigenschaften dieser Person oder dieses Objekts. Das Wort *record* hat sich deshalb zur Beschreibung eines Verbunds von Daten dieser Art eingebürgert.
Die Record- und die Array-Strukturen haben die gemeinsame Eigenschaft, dass beide beliebig zugreifbare Strukturen sind. Der Record ist insofern allgemeiner, als nicht alle Komponententypen identisch sein müssen. Der Array seinerseits bietet größere Flexibilität, da die Komponenten-Selektoren berechenbare Werte sein können (dargestellt durch Ausdrücke), während die Selektoren von Record-Komponenten festgelegte, in der Record-Typendefinition enthaltene Namen sind. Eine wichtige Rolle spielen Records in Datenbanken [Lockemann et al. 1987].

Menge

Eine Menge ist eine (zunächst) ungeordnete Struktur.
Die Strukturart *Set* wird nach folgendem Schema definiert:

```
TYPE T = SET OF T0
```

Die möglichen Werte einer Variablen x vom Typ T sind Mengen von Elementen von T0. Die Menge aller Untermengen von Elementen einer Menge T0 wird die Potenzmenge von T0 genannt. Der Typ T umfaßt somit die Potenzmenge seines Grundtyps T0.

Die unmittelbare Folge dieser Beschränkung ist, dass Elemente nicht direkt zugreifbar sind, mit der Ausnahme des einen Elementes, das gerade an der Reihe ist. Es ist dieses Zugriffsprinzip, das die Sequenz grundsätzlich vom Array unterscheidet.

Array, Record und Set sind Random-Access-Strukturen. Sie werden für Daten verwendet, die sich im Hauptspeicher befinden, der beliebigen Komponenten-Zugriff erlaubt.
Rekursion bewirkt den Übergang zu abzählbar unendlichen Wertebereichen, deren Elemente gleichartig aus einfacheren Elementen des Wertebereichs aufgebaut sind. In imperativen Programmiersprachen werden rekursive Datenstrukturen durch Zeiger gebildet (sog. Geflechte). Beispiele für rekursive Datenstrukturen sind lineare Listen und Bäume.

Liste

Eine Liste ist eine verkettete Folge von Elementen eines gegebenen Datentyps. Man unterscheidet zwischen einfach verketteten und doppelt verketteten linearen Listen. Verbindungen zwischen den Elementen einer Liste werden am einfachsten dadurch hergestellt, dass man sie in einer einzigen Liste aufreiht. Dann bedarf es einer einzigen Zeigervariablen pro Element, welche die Verbindung mit dem jeweiligen Nachfolger herstellt.
Bei der Generierung einer Liste geht man so vor, dass – ausgehend von der leeren Liste – immer ein neues Kopfelement angefügt wird. Bei einer Liste hat jedes Element einen Vorgänger und einen Nachfolger. Ausnahmen bilden das erste (kein Vorgänger) und das letzte Element (kein Nachfolger). Damit kennt jedes Element seinen direkten Nachfolger; mit Ausnahme des letzten Elements, das ja keinen Nachfolger hat. Also hat jedes Element implizit Kenntnis darüber, ob es das

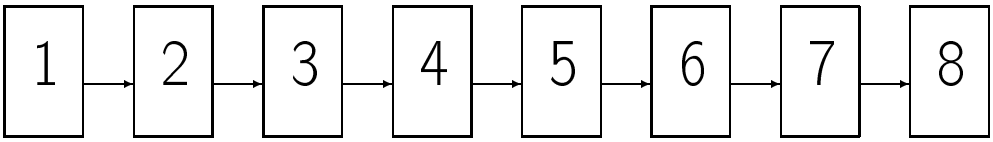


Abbildung 5.1: Eine Liste in eindimensionaler Darstellung

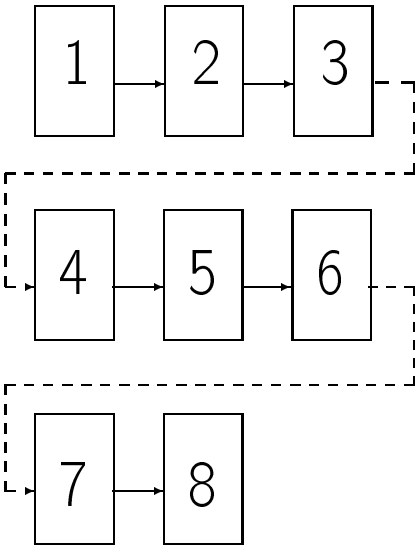


Abbildung 5.2: Die obige Liste in zweidimensionaler Darstellung (wie auf einem Bildschirm), obwohl die Liste von ihrer Struktur her noch immer eindimensional ist

letzte Element in der Liste ist oder nicht. Kein Element hat aber die Information darüber, ob es das erste Element in der Liste ist.
Eine Liste ist linear, auch wenn sie auf dem Bildschirm in den 2D-Raum abgebildet werden kann. Die Linie, die die Liste bildet, wird dazu gekrümmt. Es bedarf aber einer Hilfslinie, um immer an den Anfang der Zeile gelangen zu können. Siehe dazu auch die Abbildungen 5.1 und 5.2.
Der entscheidende Vorteil verketteter Listen gegenüber Feldern besteht darin, dass ihre Größe zu- und abnehmen kann, solange sie existieren; insbesondere muss ihre maximale Größe nicht im voraus bekannt sein. In praktischen Anwendungen wird es dadurch oft möglich, dass sich verschiedene Datenstrukturen den gleichen Speicherbereich teilen, ohne dass man zu irgendeinem Zeitpunkt speziell auf ihre jeweiligen Größen achten muss. Eine verkettete Liste ist eine Menge von

Elementen, die – wie in einem Feld – sequentiell organisiert sind. In einem Feld wird die sequentielle Anordnung implizit bewirkt (durch die Position im Feld); in einer verketteten Liste verwendet man eine explizite Anordnung, in welcher jedes Element Teil eines Knotens ist, der auch eine Verkettung zum nächsten Knoten enthält.
Möchte man in einem Feld das letzte Element an den Anfang setzen, so müsste man jedes Element verschieben, um Platz für das neue Element am Anfang zu schaffen. In einer verketteten Liste verändert man nur drei Verkettungen (egal wie lang die Liste ist).
Zum „Finden des k-ten Elements“ (Auffinden eines Elements, wenn sein Index gegeben ist) sind Listen nicht gut geeignet, da k Verkettungen durchlaufen werden müssen. In einem Feld wird dies einfach durch den Zugriff $a[k]$ realisiert. Ebenso ist das „Finden des Elements *vor* einem gegebenen Element“ schwierig. Dazu muss die gesamte Liste durchgegangen werden, um den Knoten zu finden, der auf das gegebene Element zeigt.
Zyklische Listen geben einem Programm die Möglichkeit, die Liste immer wieder zu durchlaufen, solange noch etwas in ihr enthalten ist. Man lässt den letzten Knoten auf den ersten zeigen. Die Struktur, die man so erhält, ist der Ring.

Keller

Die wichtigste Datenstruktur mit beschränktem Zugriff ist der Keller, Stack oder Stapel. Nur zwei Grundoperationen treten auf: Man kann ein Element auf dem Stapel ablegen (*push*), und man kann ein Element entnehmen (*pop*). Es ist typisch, dass für eine große Zahl von Operationen nur ein kleiner Stapel benötigt wird. Wenn man sicher ist, dass dies der Fall ist, so kann eine Felddarstellung verwendet werden. Andernfalls kann eine verkettete Liste dem Stapel die Möglichkeit geben, sich auf elegante Weise zu vergrößern und zu verkleinern. Bei der Implementierung eines Kellers wird man in der Regel noch die Funktionen *top* (liefert das oberste Element ohne den Stack zu verändern) und *empty* (prüft, ob der Keller leer ist) ergänzen. Ein Keller ist eine LIFO-Liste (Last In First Out).

Schlange

Eine weitere grundlegende Datenstruktur mit beschränktem Zugriff wird Schlange oder Queue genannt.
Elemente dürfen nur am Ende der Schlange eingefügt und am Anfang entfernt werden.
Formal geschrieben: Gegeben sei ein Datentyp T. Elemente aus T werden in der Schlange als Folge organisiert, und es gibt zwei Zugriffsoperationen, von denen die

eine ein Element aus T stets an das Ende der Folge anfügt und die andere stets das erste Element der Folge entfernt und als Ergebnis liefert. Die Einfügeoperation wird oft mit *enter*, die Ausfügeoperation mit *remove* bezeichnet.
Das Prinzip, dass stets das zuerst eingefügte Element einer Schlange als erstes wieder ausgelesen werden muss, bezeichnet man als FIFO-Prinzip (First In First Out). Eine Schlange heißt daher auch FIFO-Speicher.
In der Praxis implementiert man Schlangen in der Regel als lineare Listen. In Programmiersprachen, die die Implementierung von linearen Listen nicht gestatten, oder bei längenbeschränkten Schlangen kann man eine Schlange auch als Feld implementieren.

Die bisher behandelten Strukturen sind ihrem Wesen nach eindimensional: Ein Element folgt dem anderen. Im nachfolgenden betrachte ich zweidimensionale verkettete Strukturen: Bäume.

Baum

Ein Baum ist eine nichtleere Menge von Knoten und Kanten, die gewissen Forderungen genügt. Ein Knoten ist ein einfaches Objekt, das einen Namen haben und andere mit ihm verknüpfte Informationen tragen kann; eine Kante ist eine Verbindung zwischen zwei Knoten. Ein Pfad in einem Baum ist eine Liste von unterschiedlichen Knoten, in welcher aufeinanderfolgende Knoten durch Kanten im Baum verbunden sind. Einer der Knoten im Baum wird als die Wurzel bezeichnet; die für die Definition entscheidende Eigenschaft eines Baumes ist, dass es zwischen der Wurzel und jedem beliebigen anderen Knoten im Baum genau einen Pfad gibt.
Jeder Knoten (außer der Wurzel) besitzt genau einen Knoten, der sich unmittelbar über ihm befindet und als sein direkter Vorgänger (parent) bezeichnet wird; die Knoten unmittelbar unter einem Knoten werden seine direkten Nachfolger (children) genannt.
Zusammenhang zwischen Listen und Bäumen:

Ein geordneter Baum ist ein Baum, dessen Verzweigungen in jedem Knoten geordnet sind. Ein Knoten y, der direkt unter einem Knoten x liegt, heißt (direkter) Nachfolger von x; ist x auf der Stufe i, so ordnet man y die Stufe i+1 zu. Umgekehrt heißt der Knoten x direkter Vorgänger von y. Die Wurzel eines Baumes liegt nach Definition auf Stufe 0. Die größte Stufe eines Elementes des Baumes heißt seine Höhe. Hat ein Element keine Nachfolger, so heißt es Endelement oder Blatt; ein Element, das nicht Endelement ist, wird innerer Knoten genannt. Die

Zahl der direkten Nachfolger eines inneren Knotens ist sein Grad. Der höchste Grad unter allen Knoten ist der Grad des Baumes. Die Zahl der Kanten von der Wurzel bis zu einem Knoten heißt Weglänge. Allgemein hat ein Knoten auf Stufe i die Weglänge i. Die Weglänge eines Baumes ist definiert als die Summe der Weglängen aller seiner Knoten. Sie heißt auch innere Weglänge.
Es gibt im wesentlichen drei Ordnungen, die sich aus der Baumstruktur auf natürliche Art ergeben. Ist W die Wurzel und bilden A und B den linken (A) und rechten (B) Teilbaum, so ergeben sich die drei Ordnungen

- Preorder: W, A, B ; besuche erst die Wurzel, dann den linken und rechten Teilbaum. Daraus ergibt sich Prefix-Notaton.
- Inorder: A, W, B ; besuche erst den linken Teilbaum, dann die Wurzel und den rechten Teilbaum. Es ergibt die Infix-Notation (die gebräuchlichste Notation).
- Postorder: A, B, W ; besuche erst den linken Teilbaum, dann den rechten und zum Schluss die Wurzel. Dies ist die Postfix-Notation.

Ein Baum ist genau dann ausgeglichen, wenn sich für jeden Knoten die Höhen der zugehörigen Teilbäume um höchstens 1 unterscheiden. Bäume, die diese Bedingung erfüllen, heißen AVL-Bäume (nach Adelson-Velskii und Landis) oder einfach ausgeglichene Bäume.

Binärer Baum

Bäume vom Grad 2 heißen binär. Sie sind entweder leer oder bestehen aus der Wurzel und einem linken und einem rechten binären Teilbaum. Bäume vom Grad größer als 2 heißen Vielweg-Bäume.
Falls jeder Knoten eine bestimmte Anzahl von direkten Nachfolgern haben muss, die in einer bestimmten Reihenfolge erscheinen, so handelt es sich um einen n-ären Baum. In einem solchen Baum ist es zweckmäßig, spezielle äußere Knoten zu definieren, die keine Nachfolger haben (und auch keine Informationen beinhalten). Äußere Knoten spielen dann die Rolle von „Pseudoknoten“, als Bezeichnung für Knoten, die nicht die vorgeschriebene Anzahl von Nachfolgern besitzen. Insbesondere ist der einfachste Typ eines n-ären Baumes der binäre Baum.
Es handelt sich um einen geordneten Baum, der aus zwei Typen von Knoten besteht: äußeren Knoten (ohne Nachfolger) und inneren Knoten mit genau zwei direkten Nachfolgern. Da die zwei direkten Nachfolger eines jeden inneren Knotens geordnet sind, spricht man vom linken und rechten Nachfolger innerer Knoten. Jeder innere Knoten muss sowohl einen linken als auch einen rechten Nachfolger

haben, obwohl es sich bei einem oder beiden von ihnen um einen äußeren Knoten handeln kann. Der Zweck des binären Baumes besteht in der Strukturierung der inneren Knoten; die äußeren Knoten dienen lediglich als Platzhalter.
Die Darstellung binärer Bäume mit zwei Verkettungen erlaubt es, am Baum abwärts zu gehen, doch man hat keine Möglichkeit, sich am Baum aufwärts zu bewegen. Die Situation entspricht der Gegenüberstellung von einfach verketteten Listen und doppelt verketteten Listen: Man kann jedem Knoten eine weitere Verkettung hinzufügen, um mehr Bewegungsfreiheit zu ermöglichen, jedoch auf Kosten einer komplizierten Darstellung.
Binärbäume lassen sich danach klassifizieren, ob die Zugriffswahrscheinlichkeiten zu ihren Elementen bei der Konstruktion und Wartung der Struktur (gewichtete Binärbäume) eine Rolle spielen oder ob für alle Elemente die gleiche Zugriffswahrscheinlichkeit angesetzt wird (nicht-gewichtete Binärbäume). Bei den nicht-gewichteten Binärbäumen muss vor allem eine in gewissen Schranken ausgewogene Höhe des Baumes garantiert werden, damit eine Degenerierung zu einer linearen Liste vermieden wird. Die bekanntesten Vertreter sind höhenbalancierte Bäume wie der AVL-Baum.

Zusammenhänge zwischen Metadatenstrukturen

Die Ähnlichkeit zwischen den Methoden zur Strukturierung von Algorithmen und denen zur Strukturierung von Daten ist groß. Die elementare, unstrukturierte Anweisung ist die Wert-Zuweisung. Das entsprechende Mitglied der Familie der Datenstrukturen ist der skalare, unstrukturierte Typ. Diese beiden sind die unteilbaren Bausteine zusammengesetzter Anweisungen bzw. Datentypen. Die einfachsten Strukturen, die man durch Aufzählung oder Aneinanderreihung dieser elementaren Bausteine erhält, sind die zusammengesetzte Anweisung und die Record-Struktur. Beide bestehen aus einer endlichen, meist kleinen Zahl von explizit aufgeführten Komponenten, die untereinander verschieden sein können. Wenn aber sämtliche Komponeneten vom gleichen Typ sind, brauchen sie nicht einzeln ausgeschrieben zu werden: In diesem Fall verwendet man die *for*-Anweisung bzw. die *Array*-Struktur, um die Wiederholung mit einem bekannten, festen Faktor anzugeben. Eine Wahl zwischen zwei oder mehr Varianten wird durch die bedingte oder die *case*-Anweisung ausgedrückt bzw. durch die variante *Record*-Struktur. Ist die Anzahl der Wiederholungen anfänglich unbekannt, so wird die Repetition durch die *while*- oder die *repeat*-Anweisung ausgedrückt. Die entsprechende Datenstruktur ist die Sequenz (file), die einfachste Art der Konstruktion von Typen mit unendlicher Kardinalität.

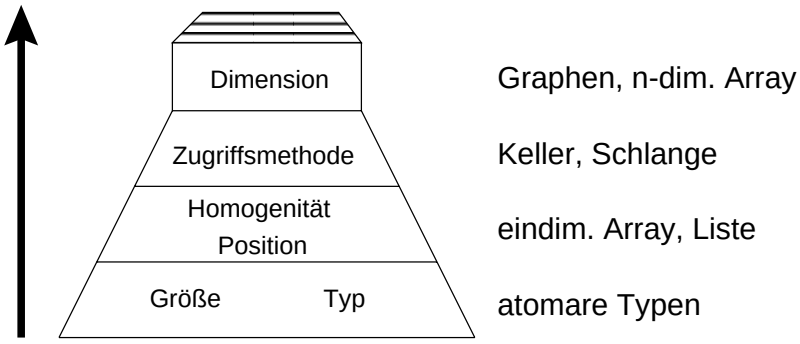


Abbildung 5.3: Zunahme der Menge an Metainformationen bei Datenstrukturen

Die Analogien zwischen Programm- und Datenstrukturen sind im folgenden noch einmal zusammengefasst [Wirth 1986]:

Konstruktions-Muster	**Programm-Anweisung**	**Daten-Typ**
atomares Element	Zuweisung	skalarer Typ
Aufzählung	zusammengesetzte Anweisung	Record-Typ
Wiederholung (Anzahl bekannt)	For-Anweisung	Array-Typ
Auswahl	bedingte Anweisung	varianter Record
Wiederholung (Anzahl unbekannt)	While- oder Repeat-Anweisung	Sequenz-Typ
Rekursion	Prozedur-Anweisung	rekursiver Datentyp
allgemeiner Graph	Sprung-Anweisung	verkettete Struktur

Ein atomarer Datentyp besitzt die Metainformationen Größe und Typ. Bei eindimensionalen Arrays und Listen kommt noch die Information über die Position hinzu. Keller und Schlangen haben alle bisher genannten Informationen und außerdem die Zugriffsmethode. Alle bisher genannten Strukturen sind eindimensional. Graphen, zu denen auch Bäume gehören sowie n-dimensionale Arrays, sind mehrdimensional Sie haben deshalb neben allen bisher genannten Metainformationen noch Kenntnis über die Dimension. In Abbildung 5.3 sollen diese Zusammenhänge noch einmal verdeutlicht werden.

Kapitel 6

Visualisierung von Informationen

Der Begriff *Information Visualization* wurde Anfang der 90er Jahre mit der Entwicklung neuer visueller Metaphern für Informationsräume am XEROX-PARC (Palo Alto Research Center) geprägt.
Unter dem Begriff Information wird i.a. das Wissen verstanden, das für die Lösung einer Aufgabe notwendig ist. Informationen müssen dazu aus einem Datenbestand herausgesucht und anschließend abgeleitet werden. Um diesen Vorgang zu erleichtern, können Daten - mit Hilfe unterschiedlicher Techniken - visualisiert werden. Da ich in meiner Arbeit eher auf neue Aspekte der Visualisierungsmöglichkeiten eingehen werde, habe ich konventionelle Visualisierungsformen wie Diagramme und Tabellen aller Art ausgelassen.
Es wird von mir beispielhaft aufgezeigt, welche Visualisierungsformen sich für welche Art von Daten eignen. Die Art der Daten ist davon abhängig, wie die Daten vor der Speicherung strukturiert wurden.

6.1 Datenstrukturen für die Informations-Visualisierung

Bei der Informations-Visualisierung kann zwischen folgenden Strukturen unterschieden werden (siehe [Shneiderman 1998, Seite 535] oder [Behme et al. 1998, Seite 43] oder auch [Morse 1998, Seite 1]):

- Eindimensionale Datenstrukturen,
- zweidimensionale Datenstrukturen,
- dreidimensionale Datenstrukturen,

- temporale Datenstrukturen,
- mulitdimensionale Datenstrukturen,
- Baumstrukturen,
- Netzwerkstrukturen.

6.1.1 Eindimensionale Datenstrukturen

Eindimensionale oder lineare Datenstrukturen umfassen geordnete Listen, Dokumente oder auch Programmcode, also alle sequentiellen Organisationformen von Information. Als anschauliches Beispiel kann man sich ein Textfenster vorstellen, das zeilenweise dargestellt wird. Systeme, die auf dieser Datenstruktur aufbauen, sind z.B. *SeeSoft* oder die *Perspective Wall*:

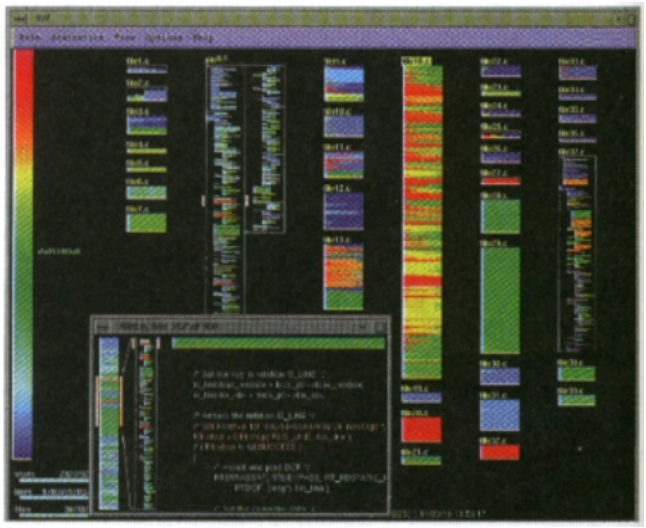


Abbildung 6.1: SeeSoft (siehe auch Farbtafel 2)

- *SeeSoft* [Eick et al. 1992] ist ein Programm zur Verwaltung von Software-Projekten und soll es den Benutzern – in diesem speziellen Fall Software-Entwicklern – erleichtern, Fragestellungen wie: „Wer hat welchen Code, wann und wozu geändert?“ beantworten zu können. Dies ist insbesondere in größeren Projekten wichtig, wo viele verschiedene Entwickler in einem Projekt arbeiten. Die Visualisierung der einzelnen Projektdateien erfolgt

über sog. „Data Visualization Sliders“. Die Länge der einzelnen Codezeilen einer Datei wird durch die Länge der senkrechten Balken in den „Sliders“ dargestellt. Jede Pixelzeile entspricht in etwa einer Codezeile und erhält eine Farbe. Da es für den Benutzer aber sehr schwer ist, aus diesem „Farbgewimmel“ Schlussfolgerungen zu ziehen, können Bereiche extrahiert, selektiert und neu eingefärbt werden. Beim Hineinzoomen in ein Textdarstellungsfenster werden die Zeilen in der derzeit gültigen Farbe dargestellt und es können Änderungen am Code vorgenommen werden. Abbildung 6.1 zeigt ein Beispiel zu dieser Anwendung.

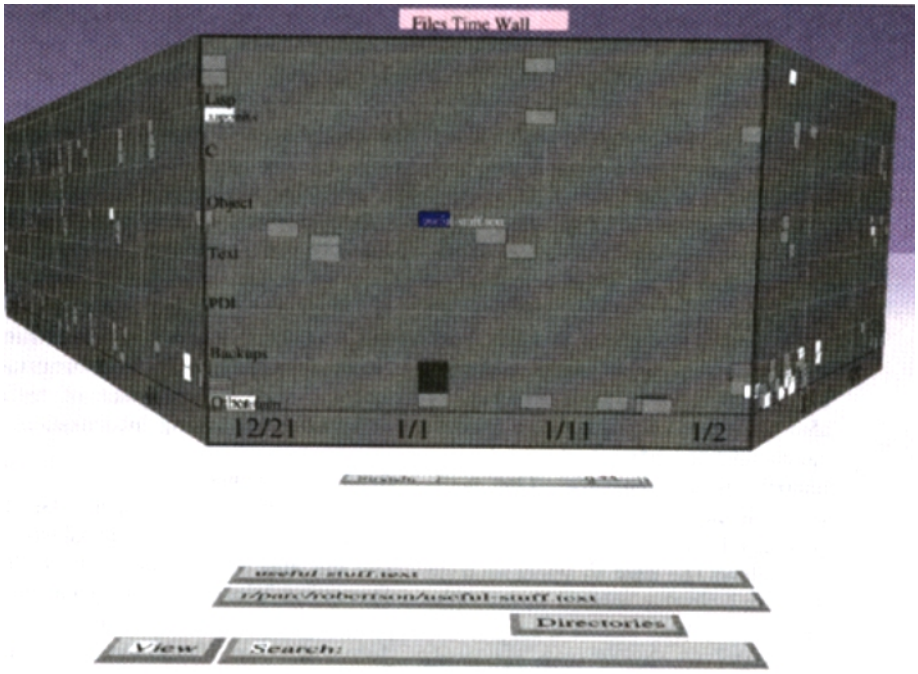


Abbildung 6.2: Perspective Wall

- Mit der *Perspective Wall* [Robertson et al. 1993a] werden u.a. lineare Strukturen, die in chronologischer oder alphabetischer Reihenfolge angeordnet sind, visualisiert. Durch die Darstellung, bei der eine virtuelle Wand dreidimensional aufgespannt wird, ergibt sich eine effektivere Ausnutzung des Bildschirmplatzes. Im Vordergrund werden die Detailinformationen angezeigt, an den Seiten werden die Informationen nach hinten hin immer weniger detailliert. Diese Darstellung vermittelt eine intuitive Fisheye-Ansicht (siehe dazu Abbildung 6.2). Die Karten, die sich an den Wänden befinden,

repräsentieren Datensätze. Für den Benutzer soll es möglich sein, Korrelationen zwischen den Karten erkennen zu können. Deshalb wird durch Lage, Größe und Farbgebung zusätzliche Information geliefert. Die *Perspective Wall* ist ein interaktives System. Wählt der Benutzer eine bestimmte Karte aus, so vergrößert sich das Element und erscheint in der Mitte der Wand, was eine Verschiebung der gesamten Wand zur Folge hat.

Die *Perspective Wall* ist besonders gut für die Darstellung von linearen Daten, die in Bezug auf eine Dimension sortiert sind, geeignet.

6.1.2 Zweidimensionale Datenstrukturen

Zweidimensionale oder planare Datenstrukturen umfassen Diagramme, kartographische Daten, Grundrisse (eine sehr ausführliche Beschreibung der Visualisierungsmöglichkeiten hierzu findet sich in [Bertin 1974]) oder Bilder und Grafiken im allgemeinen, also alle Informationsstrukturen, deren Elemente sich in einer Ebene anordnen lassen. Diese Datenstruktur hat deshalb eine herausragende Bedeutung, weil wir hauptsächlich flache Medien wie Bildschirme oder Papierseiten zur Abbildung der Realwelt benutzen.

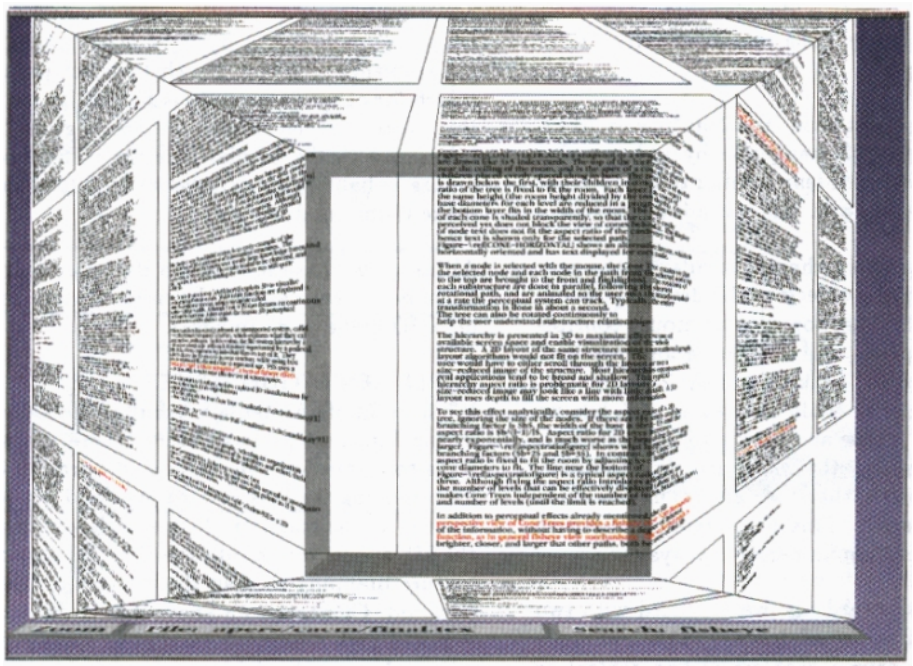


Abbildung 6.3: Document Lens

Document Lens[Robertson et al. 1993b] eignet sich für die Darstellung zweidimensionaler Datenstrukturen. Oftmals besteht diese Struktur aus Dokumentseiten, die dann in einer rechteckigen Fläche ausgebreitet werden. Im Fokus wird der Inhalt detailliert wie unter einer Lupe oder Linse dargestellt und kann betrachtet bzw. gelesen werden. Alles andere erstreckt sich kontinuierlich räumlich in die Tiefe. Der Benutzer erhält auf diese Weise eine Fisheye-Sicht auf eine abgestumpfte Pyramidenform, die den Ausschnitt vorne im Detail und an den vier trapezartigen Seiten den Kontext verzerrt darstellt (siehe Abbildung 6.3). Die im Beispiel verwendete Textdatei ist zur Darstellung eher ungeeignet, da Text eindimensional ist. Eine bessere Möglichkeit ist die Verwendung von Grafiken, die zweidimensional sind.
Die Linse lässt sich vom Benutzer nach Belieben bewegen, es ist damit allerdings auch immer ein Standortwechsel verbunden, um den Ausschnitt stets im Bildschirmrahmen zu belassen.

6.1.3 Dreidimensionale Datenstrukturen

Dreidimensionale Raumstrukturen werden traditionell in der wissenschaftlichen Visualisierung eingesetzt, um die physikalische Erfahrungswelt mit Volumenobjekten, räumlicher Interaktion und komplexen Objektrelationen zu simulieren. Ein sinnvoller Einsatz ist aber nur möglich, wenn die räumlichen Dimensionen auch sichtbar gemacht werden können.
Bei der Verwendung von zweidimensionalen Ausgabegeräten, wie z.B. Bildschirmen, werden Verfahren wie Perspektive, Beleuchtung, Schattierung und Animation verwendet, um eine räumliche Tiefe zu erschaffen. Eine weitere Möglichkeit bieten Virtual Reality-Technologien. Sie machen sich stereoskopische Verfahren zu nutze, um das räumliche Sehen zu ermöglichen. In diesen Bereich fallen auch raumbezogene Navigationsmöglichkeiten wie fly-over, fly-through usw., die zum Explorieren einer virtuellen Welt verwendet werden können.
Die 3D-Darstellung macht es aber auch Architekten möglich, sich von ihren Entwürfen ein „richtiges“ Bild machen zu können, indem sie in ihrem Entwurf „herumspazieren“ können. In der realen Welt sind viele Dinge dreidimensional, wie z.B. der menschliche Körper, Häuser und Moleküle und die drei Dimensionen sollten, weil alle wichtig sind, auch so dargestellt werden. Systeme, die dreidimensionale Datenstrukturen visualisieren sind z.B. *WebBook* bzw. *WebForager* und *Voxel-Man*:

- *WebBook* und *WebForager* (beide in [Card et al. 1996]). Die Idee bei dieser Visualisierungsform ist es, möglichst nah an der Buchmetapher zu bleiben. So werden ganze Webseiten lose zu einem virtuellen, dreidimensionalen

Buch – dem *WebBook* (siehe Abbildung 6.4 links) – gebunden, in dem man blättern kann, das mit Lesezeichen versehen werden kann und wo auch alle oder nur bestimmte Seiten flach auf den virtuellen Boden gelegt werden können, um dann mit der virtuellen Lupe gelesen werden zu können. Der *WebForager* ordnet die *WebBooks* auf dem Bildschirm an. Das aktuell verwendete Buch befindet sich im Vordergrund, andere Bücher, die erst vor kurzem benutzt wurden, befinden sich weiter im Hintergrund und nur selten verwendete Bücher stehen im Regal an der Wand (siehe Abbildung 6.4 rechts).

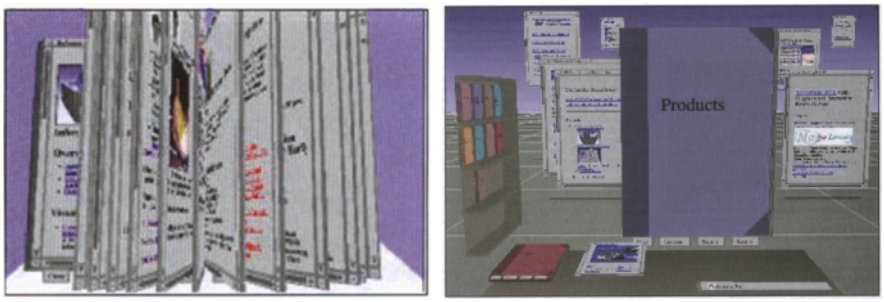


Abbildung 6.4: WebBook und WebForager

- Mit dem *Voxel-Man* [Voxel-Man 2001] hat man die Möglichkeit, sich die menschliche Anatomie dreidimensional auf dem Bildschirm anzeigen zu lassen. Dabei lassen sich die Darstellungen drehen und ermöglichen so für den Mediziner eine vollständig neue Einsicht in den Patienten. Auf diese Weise lassen sich z.B. Operationen besser planen oder Studenten können an den Modellen üben, bevor sie sich an den Patienten wagen. Abbildung 6.5 zeigt als Beispiel den menschlichen Schädel mit einem Einblick in das Gehirn .

6.1.4 Multi-dimensionale Datenstrukturen

Vor allem abstrakte Informationsobjekte sind multivariant, d.h. diese Objekte enthalten mehr als nur zwei oder drei Objektattribute. Ein abstrakter multidimensionaler Informationsraum, der in der Regel durch die Objekteigenschaften aufgespannt wird, kann zwar mathematisch n-dimensional beschrieben, aber höchstens dreidimensional visualisiert werden. Daraus ergibt sich das Problem der Projektion eines hochdimensionalen Datenraums in einen niedrigdimensionierten visuellen Informationsraum. Datenbanken sind z.B. ein gutes Beispiel für n-dimensionale Räume. Hier werden Einträge mit n Attributen zu Punkten in

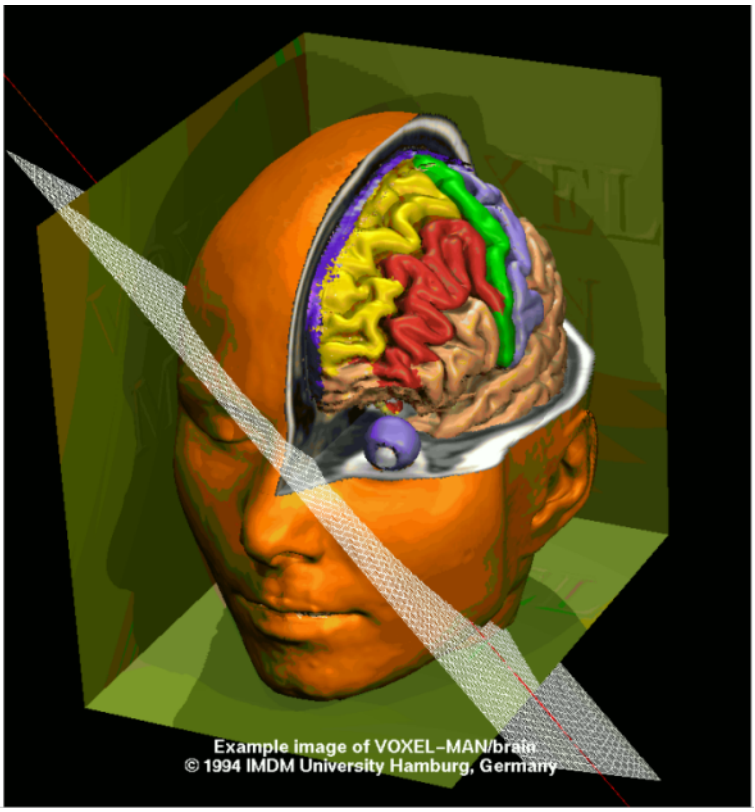


Abbildung 6.5: Voxel-Man

einem n-dimensionalen Raum. Es folgen einige Beispiele zur Visualisierung multidimensionaler Datenstrukturen:

- *HomeFinder* und *FilmFinder* (beide in [Ahlberg et al. 1994]) stellen ihre Anfrageergebnisse als farbige Punktwolken in einem abgegrenzten Bereich dar. Bei Bedarf werden diese Ergebnisse mit Hilfe von „Slidern“ dynamisch aktualisiert.

  Beim *HomeFinder* (Abbildung 6.6) soll es Haus- oder Eigentumswohnungssuchenden erleichtert werden, ihr „Traumobjekt“ zu finden. Durch Schieberegler kann z.B. die Lage, die Zimmeranzahl oder der Preis ausgewählt werden und man erhält sofort ein Feedback auf seine Anfrage, die auch ganz einfach immer wieder verändert werden kann.

  Beim *FilmFinder* (Abbildung 6.7) ist jede Farbe einem bestimmten Film-

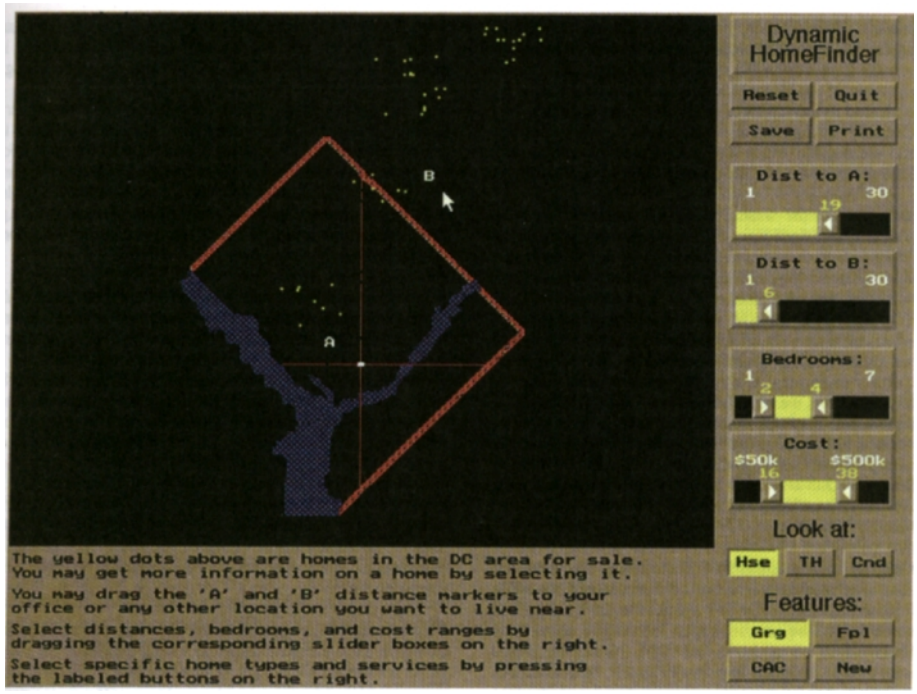


Abbildung 6.6: HomeFinder

genre zugeordnet. Die Position der Punkte ergibt sich aus seinem Bekanntheitsgrad (vertikale Ausrichtung) und seinem Alter (horizontale Ausrichtung). Mit den Schiebereglern können Titel, Schauspieler, Regisseure und die Länge des anzuzeigenden Films ausgewählt werden. Auf diese Weise kann man die Treffermenge verkleinern oder vergrößern.

- *VisDB* [Keim et al. 1994] ermöglicht die Visualisierung und interaktive Exploration von sehr großen Datenbanken, die multidimensionale und unstrukturierte Daten enthalten können. Jeder Datenwert wird durch ein farbiges Pixel dargestellt. Dadurch können theoretisch - je nach Größe des Bildschirms und der Auflösung - bis zu einer Million Datenelemente gleichzeitig auf dem Bildschirm dargestellt werden. Auf diese Weise lassen sich auch Datensätze visualisieren, die eine Anfrage nur approximativ erfüllen. Die Datensätze mit der höchsten Trefferquote werden im Zentrum positioniert, die restlichen - in absteigender Relevanz - in einer zweidimensionalen Spiral-, Schlangen- oder Gruppenanordnung aneinandergereiht. Die Farbgebung der Pixel entspricht dabei dem Abstand zum Abfragewert. Ein „Volltreffer“ er-

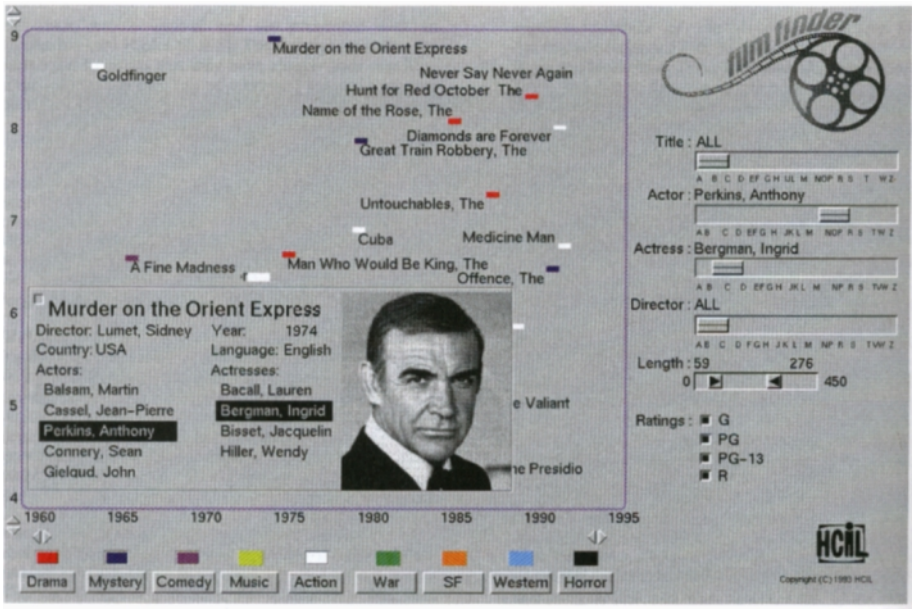


Abbildung 6.7: FilmFinder

hält die Farbe gelb, alle weiteren Pixel werden - je nach Relevanz - dunkler dargestellt. Siehe dazu auch Abbildung 6.8

- *Table Lens* [Rao et al. 1994] soll die Visualisierungsmöglichkeiten von großen Tabellen, die häufig kaum zu überblicken sind, verbessern. Hierfür wird die Fisheye-Technik angewendet, die es ermöglicht, gleichzeitig die gesamte Informationsstruktur zu zeigen und auch in einen bestimmten Punkt hineinzuzoomen ohne den Gesamtüberblick zu verlieren. Die bekannten Eigenschaften von Tabellen wie die Darstellung von Zeilen und Spalten sowie ihre Zusammenhänge finden auch hier Verwendung. Allerdings sind zusätzlich symbolische und grafische Elemente integriert. Informationen werden mittels Text, Farbe, Schattierung und Balkenlänge vermittelt. Siehe dazu auch Abbildung 6.9. Mit *Table Lens* können bis zu knapp 70.000 Datenbankeinträge auf einer Bildschirmseite angezeigt werden . Um einen genauen Wert ablesen zu können, muss der Benutzer an der gewünschten Stelle die Tabellen-Lupe (Table Lens) aufziehen, die dann eine oder mehrere Zeilen gut lesbar darstellt, den Rest – wie bisher – klein. Durch die Möglichkeit, interaktiv verschiedene Bereiche in unterschiedlichem Maß zu detaillieren, indem Zeilen und Spalten voneinander unabhängig in Höhe und Breite va-

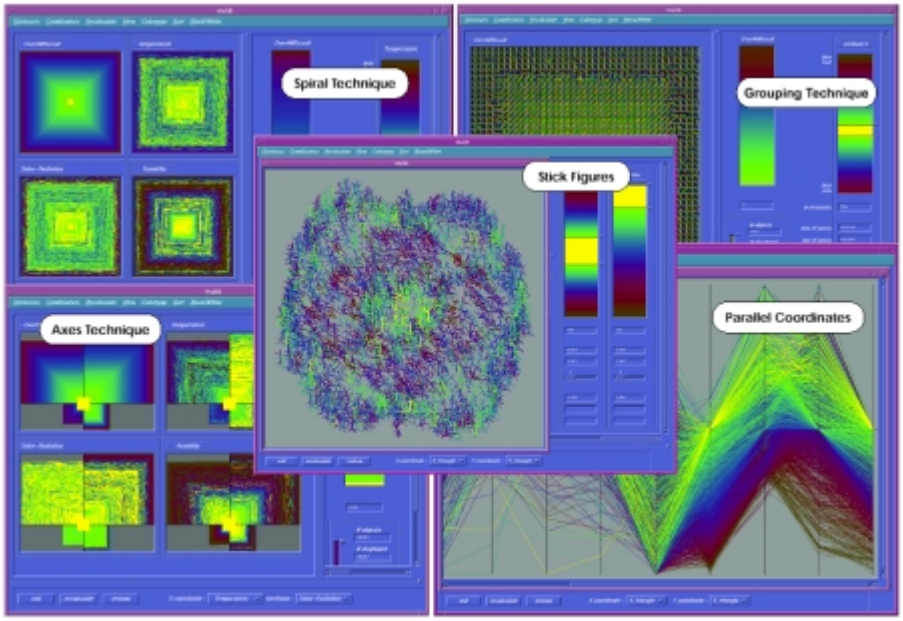


Abbildung 6.8: Verschiedene Bildschirmansichten von VisDB (siehe auch Farbtafel 3)

riiert werden, kann der Benutzer Korrelationen schnell erkennen.

Table Lens eignet sich besonders gut für die Informationsexploration aus großen relationalen Datenbeständen.

Eine Einführung in *Table Lens* mit Beispielen findet sich in [Inxight 2001]

6.1.5 Temporale Datenstrukturen

Temporale Datenstrukturen werden dort eingesetzt, wo der zeitliche Aspekt von Daten eine große Rolle spielt, also z.B. im Projektmanagement, bei medizinischen Aufzeichnungen oder beim Aufzeichnen und Abspielen von Musik und Filmen. Wichtig ist hierbei, dass die Einträge alle einen Start- und einen Endzeitpunkt haben. Trotzdem kann es zwischen den Einträgen natürlich Überlappungen geben. Ist die Visualisierung von temporalen Datenstrukturen gut gelungen, so lassen sich Antworten auf Fragen wie: „Welche Ereignisse waren vor, während oder nach einer bestimmten Zeitperiode“ schnell finden.
Zeitabhängige Informationen sind ein spezieller Typ von sequentiellen Datenstrukturen. Deshalb finden auch die Systeme, die die linearen Datenstrukturen

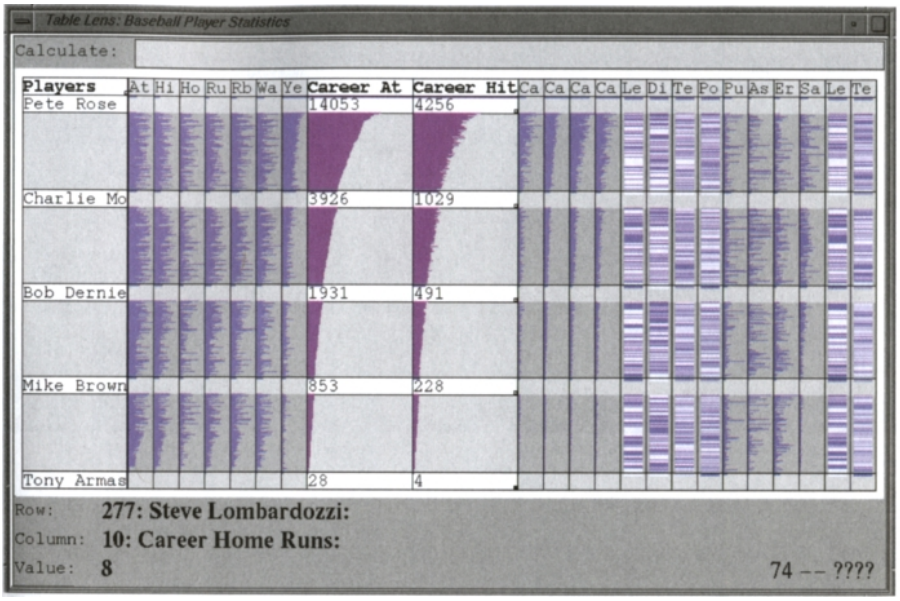


Abbildung 6.9: Table Lens

visualisieren, hier wieder Verwendung. Allerdings muss der zeitliche Aspekt als ein zusätzliches Kriterium aufgenommen werden. So wird z.B. die *Perspective Wall* zur *Time-Wall* (Abbildung 6.2), die Dokumente entsprechend der zeitlichen Reihenfolge auf der Oberfläche der perspektivischen Wand anordnet.

6.1.6 Baumstrukturen

Typische Beispiele für Baum- oder Hierarchiestrukturen sind Inhaltsverzeichnisse in Büchern oder Dateisysteme in der Computertechnik (alle gängigen Betriebssysteme unter `Windows`, `UNIX` oder `MacOS` bieten die Visualisierung der Dateistrukturen als Dateibaum an). Hierarchische Informationsräume können sehr effizient automatisch generiert werden, falls die Basisinformationen bereits strukturiert sind. Eine ausführliche Beschreibung zu Thema Bäume findet sich auf Seite 99. Liegen die Daten als Hierarchie oder Baumstruktur vor, so sind folgende Punkte von Nachteil:

- Baumstrukturen sind für die Darstellung sehr großer Datenmengen ungeeignet.
- Mit dem Zwang der Einordnung jedes Elements in eine statische Klassifikationsstruktur werden bestimmte Objektverknüpfungen entweder verdeckt

oder man muss redundante Informationen in die Baumstruktur einfügen.

- Die Navigation in Baumstrukturen und Entscheidungsbäumen ist problematisch, falls die Hierarchie nur teilweise dargestellt werden kann, z.B. bei zweidimensionalen Darstellungen.

Viele Prototypen von Informations-Visualisierern verwenden eine hierarchische Dokumentstrukturierung, um den Benutzer bei der Informationssuche visuell zu unterstützen. Das sind z.B. die zweidimensionalen *Treemaps*, die dreidimensionalen *Cone-Trees* oder *Hyperbolic Trees*:

- Mit dem *Hyperbolic Tree* [Lamping et al. 1996] lassen sich große hierarchische Strukturen visualisieren. Die Baumhierarchie wird in einer hyperbelartigen Ebene ausgebreitet und auf eine zweidimensionale, kreisförmige Fläche projiziert (siehe Abbildung 6.10). In der Nähe des Kreisrandes wird der Baum kontinuierlich kleiner dargestellt (Fisheye-Sicht). Dadurch, dass den Knoten und ihren Nachfolgern keilförmige Segmente zugewiesen werden, ist eine Überschneidung der Kanten nicht möglich. Der Benutzer kann per Mausklick einen Teilbaum auswählen, woraufhin sich der Baum in einer fließenden Animation bewegt und der gewünschte Bereich dann vergrößert in der Mitte erscheint. Der Kontext ist stets in Form der Nachbarknoten sowie durch Vorgänger und Nachfolger sichtbar.

- Mit dem *Tree-Map*-Konzept [Johnson et al. 1991] kann man die gesamte zweidimensionale Bildschirmoberfläche ausnutzen und eine große hierarchische multivariante Datenmenge mit über 1000 Knoten gleichzeitig darstellen. Dazu werden die Ebenen des Baumes abwechselnd in vertikale und horizontale rechteckige Regionen aufgeteilt und so platziert, dass sie die logische Baustruktur widerspiegeln. Tiefere Ebenen befinden sich innerhalb ihrer Vorgänger. Das sich ergebende Muster erinnert an ein Mosaik, bei der jede Kachel einen Knoten darstellt. Durch Farbe und Beschriftung wird für den Anwender zusätzlich Information geliefert. Die Anwendung ist interaktiv und erlaubt es dem Benutzer z. B. Datenparameter zu gewichten oder in ausgewählte Bereiche zu zoomen.

  Die Methode ist gut geeignet, um einen Überblick über eine große Anzahl hierarchisch geordneten Informationen zu liefern. Denkt man z.B. an einen Windows-Dateibaum, in dem sehr viele Verzeichnisse und Unterverzeichnisse vorhanden sind, dann kann man diesen Baum sehr schnell nicht mehr auf einer Bildschirmseite darstellen und der Benutzer muss sehr viel „herumscrollen“, um alles sehen zu können. Bei einer Darstellung als Tree-Map

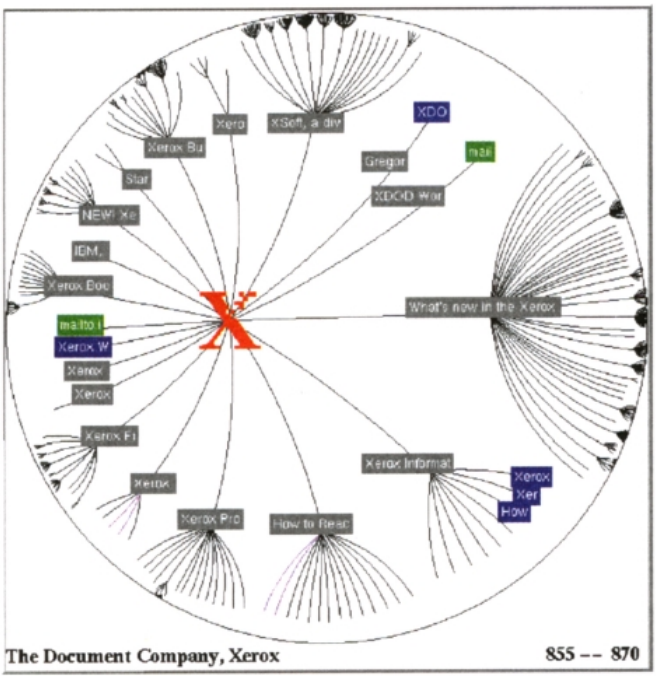


Abbildung 6.10: Hyperbolic Tree

hätte der Benutzer die gesamte Verzeichnisstruktur auf einen Blick. Allerdings kann die fehlende explizite Strukturdarstellung zu einem Kontextverlust und einer Desorientierung führen. Außerdem kann es schwierig sein, die Hierarchieebenen der einzelnen Regionen richtig auseinanderzuhalten.

- Beim *Cone-Tree* [Robertson et al. 1991] erfolgt, ähnlich wie bei zweidimensionalen Bäumen, eine von der Wurzel (Kegelspitze) ausgehende Verfeinerung des dargestellten hierarchischen Sachverhaltes in die Tiefe. In den Blättern (Kegelbasen) befinden sich somit die detailliertesten Informationen. Durch die zusätzliche Verwendung der dritten Dimension wird es möglich, die Bäume in ihrer typischen Kegelform anzuordnen. Einem Baum vergleichbar handelt es sich um einen gerichteten, azyklischen Graphen. Bei der Projektion der dreidimensionalen Kegel auf den Bildschirm werden Knoten und Kanten teilweise überlagert. Für die Interaktion stehen dem Benutzer eine Reihe von unterschiedlichen Operationen zur Verfügung. So kann er einzelne Blätter entfernen oder anhängen und dadurch die Zahl der dargestellten Knoten regulieren; die Kegel lassen sich partiell visualisieren oder dynamisch umstrukturieren.

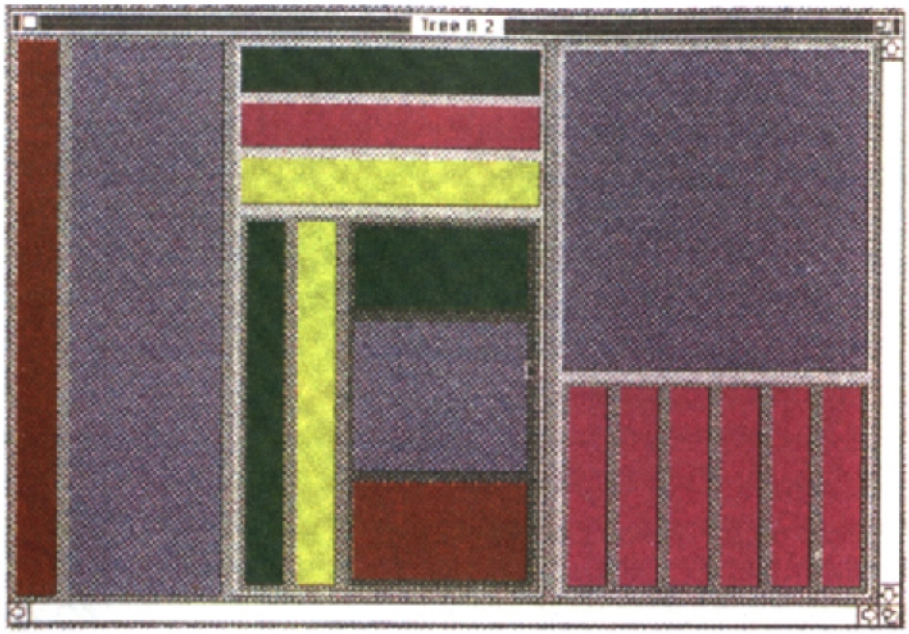


Abbildung 6.11: Tree-Map

Dem allgemeinen Problem der Verdeckung von Knoten im 3D-Raum wird durch die halbtransparente Darstellung der Knoten und durch die Möglichkeit des interaktiven Drehens des Baumes entgegengewirkt. Ein *Cone-Tree* präsentiert die Hierarchie von oben nach unten. Ein *Camtree* unterscheidet sich von einem Cone-Tree nur darin, dass bei ihm die Hierarchie von unten nach oben aufgebaut wird. Eine weitere Variante lässt den Cone-Tree horizontal von links nach rechts wachsen. Bei dieser Art von Bäumen kann man leicht für jeden Knoten einen Namen eintragen. Dies liegt daran, dass die meisten Hierarchien vom Verhältnis her eber breit als tief sind. Hierarchien mit bis zu 10000 Knoten können so mit Cone-/Camtrees dargestellt werden. Der Vorteil liegt in der sehr viel platzsparendereren Darstellung des Baumes gegenüber der zweidimensionalen Darstellung. Zudem kann man sehr gut hierarchische Abhängigkeiten darstellen. Außerdem erzielt man eine hohe Interaktivität. Ein Nachteil dieser Bäume ist der begrenzte Platz zur Darstellung von Informationen in den Blättern. Dieser Nachteil ist bei den Cone-Trees, die von oben nach unten wachsen, noch ausgeprägter, als bei denen, die sich von links nach rechts ausdehnen. Zwei Beispiele hierfür finden sich in Abbildung 6.12.

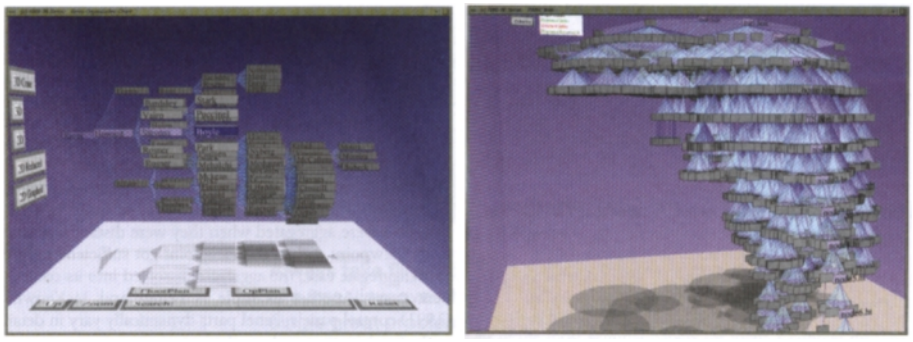


Abbildung 6.12: Cone-Tree (siehe auch Farbtafel 4)

6.1.7 Netzwerkstrukturen

Die grafische Repräsentation von Netzwerken erfolgt i.a. in der Form von Verbindungen (Links) und Knotenpunkten (Nodes). Hat jeder Knotenpunkt eine willkürliche Anzahl von Links zu anderen Konotenpunkten, kann es zu einem unüberschaubaren Bild kommen, das nur noch wie ein Wirrwar aus Verbindungen aussieht. Eine wichtige Aufgabe ist daher die Informationsfilterung, d.h. eine Vereinfachung der Netzwerkstruktur, ohne dass jedoch wichtige Elementbeziehungen verlorengehen. Bei der Visualisierung von Netzwerken kann andererseits bewußt die räumliche Struktur des Informationsraums ausgenutzt werden, um gemäß dem assoziativen Ansatz inhaltliche Nähe = räumlich Nähe Informationen im dreidimensionalen Raum darzustellen. Dieser Ansatz kann aber nur dann erfolgreich sein, wenn der Benutzer in der Lage ist, räumliche Objektabstände zu erkennen, z.B. durch den Einsatz von Virtual-Reality-Systemen.

Die dreidimensionale *Information Landscape* [Wright 1995] dient zur Darstellung von Hypermedia-Strukturen. Informationen werden durch Elemente - beispielsweise durch Sockel im Raum, auf welchen wiederum kleine Quader positioniert sind - samt den Datenattributen dargestellt. Zusätzlich werden Fläche und Höhe der Quader als auch Farbe und Beleuchtung genutzt, um dem Benutzer Informationen zu liefern. Als eine Art Informationsgeflecht werden die strukturellen Beziehungen in einer virtuellen Landschaft ausgedrückt. Der Benutzer kann auf unterschiedliche Weise mit der Information Landscape interagieren: er kann den Blickwinkel im simulierten Raum ändern, über die Ebene fliegen oder einen Bereich anvisieren und ihn im Detail betrachten. Siehe dazu auch Abbildung 6.13.

Aufgrund der ebenen Anordnung der Datenstruktur ist der Platzbedarf der Information Landscape immens. Ohne eine explizite Übersichtskarte bietet dieses

Abbildung 6.13: Information Landscape

Modell keinen Überblick über die gesamte Struktur, was die ungerichtete Suche nach Daten wenig ergiebig macht. Hohe Anforderugen werden an Rechenleistung und Speicherkapazität gestellt. Außerdem ist ein Einsatz von VR-Geräten, wie HMD und DataGlove sinnvoll, weil erst hierdurch die Möglichkeit des Eintauchens in den virtuellen Informationsraum geschaffen wird.

6.1.8 Metaphern im Informationsraum

Gute visuelle Benutzungsschnittstellen erlauben dem Benutzer von Informationssystemen, visuelle Metaphern des Informationsraums weitgehend ohne Eingewöhnungsphasen und elektronische Hilfestellungen kognitiv zu erfassen. Aus den Erfahrungen mit graphischen Windows-Point-und-Click-Oberflächen ergeben sich grundlegende Aufgaben für interaktive Retrievalschnittstellen, die von Shneiderman zusammengefaßt worden sind [Shneiderman 1996] bzw. [Shneiderman 1998, Seite 535ff]:

Überblick Einen Überblick über den gesamten Datenraum oder über einen Themenbereich bietet einen guten Startpunkt für die Informationssuche. Besonders bei großen Datenkollektionen ist es zunächst erforderlich, relevante Informationsdomänen zu finden. Problematisch ist die textliche oder symbolische Auszeichnung der Informationsdomänen bei grafischen Übersichten.

Das Fisheye-Konzept setzt beispielsweise die Überblicks-Metapher ein.

Zoom Das Zoomen ermöglicht eine visuelle Fokussierung auf wichtige Bereiche, während andere, weniger relevante Informationsstrukturen, ausgeblendet werden oder in den Hintergrund treten. Durch eine Änderung der Raumgeometrie kann erreicht werden, dass bestimmte Strukturen und Objekte vergrößert in den Vordergrund gebracht werden. Dieses Konzept ist im *Elasic Catalog* (siehe Abbildung 6.14) sehr anschaulich umgesetzt worden.

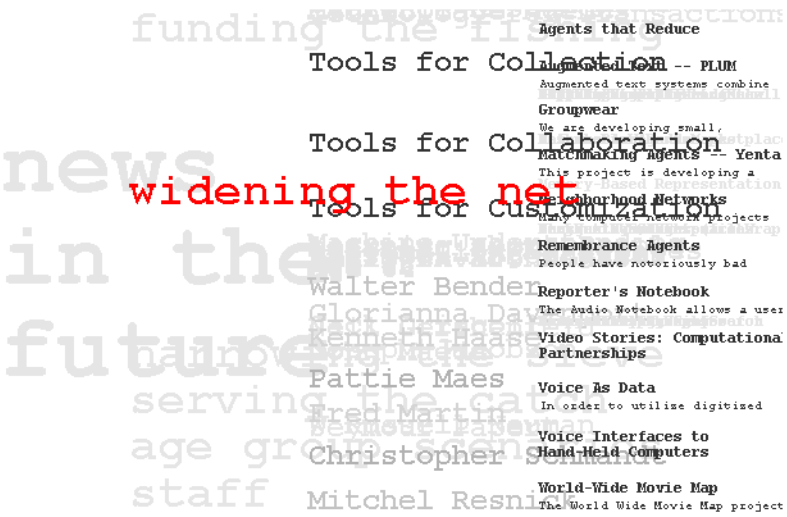


Abbildung 6.14: Elastic Catalog (siehe auch Farbtafel 5)

Filter Filter haben prinzipiell die Aufgabe, relevante von nichtrelevanten Informationen zu trennen. Bei nahezu allen visuellen Benutzungsoberflächen navigiert oder zoomt sich der Benutzer durch den Informationsraum und filtert parallel dazu relevante Informationen, indem er ihn interessierende Bereiche auswählt oder Suchpfade verfolgt.

Details auf Anforderung Hat ein Benutzer ein bestimmtes Objekt gefunden,

so muss auch gewährleistet sein, dass detaillierte und objektspezifische Daten unmittelbar zu seiner Verfügung stehen.

Objektrelationen Die Visualisierung von Objektbeziehungen ist besonders dann problematisch, wenn eine hohe Informationsdichte auftritt. Der Benutzer sollte deshalb auf jeden Fall die Art und Anzahl von Objektrelationen auswählen können. Wenig hilfreich sind Darstellungen, in denen alle abstrakten Objektrelationen einfach durch Verbindungslinien zwischen Objekten oder Knotenpunkten visualisiert werden. Dies führt oft zu einem verwirrenden Chaos von Relationen ohne semantischen Bezug und ist unter dem Begriff *Information Overload* bekannt.

History Die Speicherung der Abfolge von Benutzeraktivitäten für Rückverfolgung und Zurücknahme von Aktionen ist ein wichtiges Element zur Verfeinerung der Benutzerführung.

Extraktion Die Extraktion und Aufbereitung der im Retrievalprozess gefundenen und als relevant befundenen Informationen, einschließlich der Formatkonvertierung, Speicherung und Ausdruck, sind wichtige funktionelle Bestandteile von Retrievaloberflächen, aber bisher in nur wenigen Anwendungen zu finden (z.B. bei *Visage* [Roth et al. 1996]).

6.2 Gründe für die Verwendung von drei Dimensionen

Kriterien, die eine Anwendung dreidimensionaler Kommunikationsoberflächen sinnvoll machen:

Nachbildung der realen Welt Möchte man die Realwelt wirklichkeitsgetreu nachbilden, so ist eine Simulation dreidimensionaler Objekte und räumlicher audiovisueller Eigenschaften (Beleuchtung, Schatten, Raumklang) unumgänglich.

Verbessertes Abbildungsverhältnis Die Informationsdichte (darzustellende Informationen im Verhältnis zum verfügbaren Platz) zweidimensionaler Darstellungen (zum Beispiel von Hierarchien) wird durch den Übergang in die dritte Dimension verringert, da die Möglichkeiten zur Anordnung von Objekten um 50% gestiegen sind. Neben horizontaler oder vertikaler Anordnung ist nun auch eine Anordnung in den Raum hinein möglich.

Räumliche Bewegungsfreiheit Der Raum hebt die Einschränkung auf horizontales oder vertikales Scrollen auf und ermöglicht neue Navigationsmöglichkeiten und -erfahrungen wie fly-over, fly-through, den Zoom oder die Perspektive. Eine andere Sichtweise einzunehmen ist in dreidimensionalen Welten wortwörtlich zu verstehen.

Gesteigerte Interaktivität Dreidimensionale Informationsräume bieten ein großes Spektrum neuer audio-visueller Erfahrungen. Dazu gehören die Wechselwirkung mit Raumobjekten über Sensoren, Animationen sowie räumliche und audio-visuelle Effekte.

6.3 Layoutgestaltung

Die Layoutgestaltung virtueller Informationsräume hat einen entscheidenden Einfluss auf die Benutzerakzeptanz der gesamten Schnittstelle. Das Layoutdesign sollte i.d.R. nicht durch das Datenmodell beeinflusst werden. Vielmehr müssen in das Design vorrangig Aspekte der Navigation und Benutzerführung einfließen. Einige wichtige Layoutkonzepte für virtuelle Informationsräume wurden von Kevin Hughes [Hughes 1995] zusammengefasst:

Vertikale Fläche Die meisten fensterbasierten Umgebungen benutzen vertikale Flächen als Layout zur Anordnung von Elementen. Die Verschiebung von Objekten auf dieser Fläche und diverse Zoomfunktionen zur Detailbetrachtung gehören standardmäßig zum Funktionsumfang dieser grafischen Oberflächen.

Horizontale Ebene Flache oder räumliche Elemente sind in der Ebene angeordnet und können überflogen werden. Virtuelle Informationslandschaften werden meist auf einer horizontalen Ebene aufgebaut. Eines der beliebtesten Layouts für virtuelle Welten.

Matrix Das Matrix-Layout erweitert die vertikale Fläche dadurch, dass Objekte hier (halb-)transparent dargestellt werden. Dadurch können auch Informationen auf sonst verdeckten oder überlappenden Objekten wahrgenommen werden.

Cockpit Ähnlich der Anordnung von Instrumenten im Cockpit eines Flugzeugs werden alle Elemente um den Benutzer herum angeordnet.

Würfel Der Informationsraum wird durch drei senkrecht zueinander stehende Flächen aufgespannt. Elemente können im Raum oder auf den planaren Flächen angeordnet werden.

Fisheye Interessante Bereiche einer Fläche oder eines Raumes werden in den Vordergrund transformiert und vergrößert. Dadurch entsteht ein verzerrtes Abbild der gesamten Struktur, wobei wichtige Bereiche im Vordergrund detailliert und scharf erscheinen, momentan nicht relevante Randbereiche jedoch verzerrt und unscharf dargestellt werden.

Perspektivische Wand Ähnlich dem Fisheye-Layout; bezieht sich aber mehr auf Daten, die sequentiell organisiert werden können, also z.B. zeitbezogene Informationen.

Zylinder Elemente werden auf der Oberfläche eines Zylinders oder Kegels angeordnet. Im einfachsten Fall kann der Zylinder gedreht werden, um so die gesamte Oberfläche sichtbar zu machen. Problem: Verdeckung von Informationen.

Räumliche Layouts können nach ihrer virtuellen Informationsarchitektur klassifiziert werden. [Däßler et al. 1998, Seite 62] führen folgende Kategorien auf:

Panorama Bei der Panorama-Architektur steht der Betrachter im Zentrum der Welt. Die Informationen sind um ihn herum angeordnet und können durch ein Minimum an Bewegung (der Betrachter braucht sich nur um seinen eigenen Standpunkt zu drehen) wahrgenommen werden.

Fließende Räume Diese Räume besitzen keine explizite Grundfläche; die Objekte in ihnen sind frei schwebend und gravitationsfrei. Fließende Räume sind mit dem Weltraum vergleichbar und eignen sich besonders für die Darstellung von Scattermaps oder Netzwerkstrukturen.

Matrix Die Strukturierung abstrakter Welten (z.B. beim Information Cube) erfolgt häufig durch die Anordnung von Objekten in regulären räumlichen Abständen. Aufgrund der uniformen Strukturelemente kann jedoch die Navigation erschwert werden.

Der *Information Cube* [Rekimoto et al. 1993] (siehe Abbildung 6.15) visualisiert dreidimensional große hierarchische Informationsstrukturen. Hierarchieebenen werden durch ineinander geschachtelte Quader oder Würfel dargestellt, die überschneidungsfrei sind. Auf ihrer Oberfläche sind sie beschriftet. Durch die halbtransparente Darstellung wird erreicht, dass tiefer

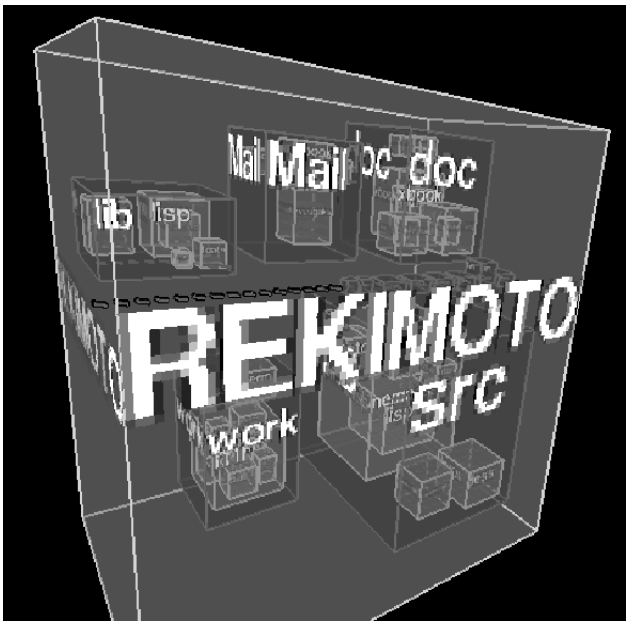


Abbildung 6.15: Der Information Cube.

liegende Ebenen sichtbar sind, aber entfernte tief liegende Ebenen verdeckt werden. Um das System benutzen zu können, benötigt man eine VR-Ausrüstung mit HMD und Datenhandschuh. Der Benutzer kann dann in einer Animation interessante Bereiche in sein Blickfeld bringen und auch einen inneren Würfel, also eine tiefere Ebene, gezielt mit einem virtuellen Strahl auswählen und die Details betrachten.

Virtuelle Landschaften Diese Informationsarchitektur wird häufig für realitätsnahe Visualisierungen genutzt (z.B. Städte- oder Gebäudearchitekturen). Sie macht sich bewusst die Gesetze der realen Welt wie Gravitation, Größenverhältnisse, Formen, Farben, Texturen, Licht und Schatten zu Nutze.

Abstrakte Landschaften Bei der Visualisierung abstrakter, multivariater Welten wird häufig von abstrakten Landschaftsarchitekturen Gebrauch gemacht. Diese haben kaum Gemeinsamkeiten mit dem real bekannten und sind im Umgang oft gewöhnungsbedürftig und wenig intuitiv. Auf der anderen Seite lässt der freie Spielraum bei der Visualisierung aber effizientere Darstellungen zu.

Probleme, die 3D-Darstellungen mit sich bringen:

- Erkennen räumlicher Abstände.
- Verdeckung von Information.
- Detailerkennung in der Raumtiefe.

Kapitel 7

Metadaten und Visualisierung

Dieses Kapitel untersucht, inwieweit Metadaten eine Hilfe bei der Auffindung einer geeigneten visuellen Präsentation sind. Zwei Wege werden im folgenden näher erörtert: Der Algorithmus von Card, Mackinlay und Shneiderman
[Card et al. 1999], der auf Relationen und die mit ihnen verknüpften Metadaten aufbaut. Der zweite Weg untersucht, inwieweit die Metainformationen in Datenstrukturen sich auf Raumkonzepte abbilden lassen, die sich direkt in eine Visualisierung ableiten lassen. Beiden Wegen ist gemeinsam, dass es keinen direkten Weg vom Datum zur Visualisierung gibt, sondern dass eine Transformation über den Zwischenschritt der Metadaten notwendig ist.

7.1 Der Algorithmus von Card, Mackinlay und Shneiderman

In [Card et al. 1999, Seite 1ff] beschreiben Card, Mackinlay und Shneiderman einen Algorithmus zur Abbildung von Daten in eine visuelle Darstellung. Grundlage hierfür bildet ein von Robertson und Ferrari entworfenes Referenzmodel für Visualisierung [Robertson et al. 1994], das in Abbildung 7.1 wiedergegeben ist.
Den Ausgangspunkt dieser Abbildung bilden die spezifischen Daten, die zunächst in eine Relation (oder eine Menge von Relationen) transformiert werden. Mathematisch formal ist eine Relation eine Menge von Tupeln $\{(Value_{ix}, Value_{iy}, \ldots), (Value_{jx}, Value_{jy}, \ldots), \ldots\}$. Für eine Visualisierung ist diese mathematische Beschreibung jedoch ungeeignet. Besser geeignet ist eine *Datentabelle*, die Relationen mit Metadaten, die die Relationen beschreiben, verbindet (siehe Tabelle 7.1).
Die Metadaten in dieser Tabelle sind die Spaltentitel ($Case_i$, $Case_j$, usw.). Daneben könnte für bestimmte Situationen auch die Reihenfolge der Zeilen ein wichti-

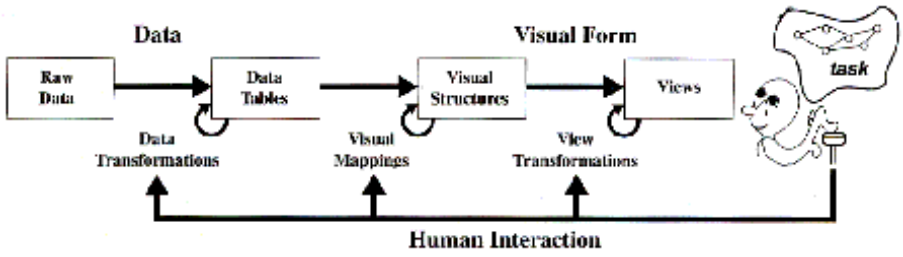


Abbildung 7.1: Referenzmodell zur Visualisierung (aus [Card et al. 1999, S. 17])

	$Case_i$	$Case_j$	$Case_k$	
$Variable_x$	$Value_{ix}$	$Value_{jx}$	$Value_{kx}$	$\cdots$
$Variable_y$	$Value_{iy}$	$Value_{jy}$	$Value_{ky}$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Tabelle 7.1: Eine Datentabelle mit Metadaten.

ges Kriterium sein, das später für eine Visualisierung benötigt wird. Die Anordnung der Reihen ist also ein weiteres Metadatum.
Statt Metadaten kommen oft auch Funktionen der Form $f(Case_i) = (Value_{ix}, Value_{iy}, \ldots)$ zum Einsatz. Da die Definitionswerte eindeutig sind, können sie in einer Tabelle als primärer Schlüssel verwendet werden (siehe Abbildung 7.2). Hierdurch wird $Case$ zu einer Variablen und ist nicht länger ein Metadatum.

Case	$Case_i$	$Case_j$	$Case_k$	
$Variable_x$	$Value_{ix}$	$Value_{jx}$	$Value_{kx}$	$\cdots$
$Variable_y$	$Value_{iy}$	$Value_{jy}$	$Value_{ky}$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Tabelle 7.2: Eine Datentabelle mit Primärschlüsseln.

Nicht jede Tabelle ist eine Datentabelle in dem hier gemeinten Sinn. Eine Distanztabelle zwischen zwei Orten, wie man sie zum Beispiel in Straßenatlanten findet, erfüllt nicht die hier gestellten Anforderungen: Über der doppelten horizontalen Linie stehen die Eingabevariablen, darunter die Ergebnisse. Dies soll aber nicht bedeuten, dass die Distanztabelle schlecht ist. Im Gegenteil, sie erfüllt gerade für den Anwender den gewünschten Zweck effizient. Die Distanztabelle müsste nun in das geforderte Format konvertiert werden. Eine Möglichkeit bestünde darin,

den Ortsnamen als Eingangsvariable zu definieren und die Längen- und Breitengerade als Ergebnis- oder Ausgangsvariablen. Aus diesen Angaben ließe sich die Distanz zwischen den Orten berechnen. Man spricht in diesem Sinne auch von abgeleiteten Variablen und Werten.
In einer Tabellen lassen sich auch hierarchische oder allgemein grafische Strukturen abbilden, wie die Tabelle 7.3 beispielhaft zeigt.

DokumentID	D_i	D_j	D_k	D_n
Bytes	4.217	5.889	1.034	12.007
Verweise	$\{D_k, D_n\}$	$\{D_i\}$	$\emptyset$	$\{D_i, D_k\}$
...	...	...	...	...

Tabelle 7.3: Datentabelle zur Beschreibung von Verweisen in einer Hypertextstruktur.

Allgemein betrachtet können Variablen in die folgenden drei Typkategorien eingeordnet werden:

- *Nominal*: Die Werte dieser Variablen können nur auf Gleichheit oder Ungleichheit geprüft werden.
- *Ordinal*: Die Werte können entsprechend einer Relation (zum Beispiel $<$) geordnet werden.
- *Quantitativ*: Auf den Werten können arithmetische Operationen durchgeführt werden.

Der Variablentyp ist entscheided dafür, auf welcher Achse später eine Visualisierung erfolgt.
Es ist möglich, die Werte einer Variablen in einen anderen Typ zu wandeln. Zum Beispiel kann eine quantitive Zeitangabe in einen ordinalen Typ gewandelt werden, indem man Zeitbereiche (kurz, mittel, lang) definiert und die Werte enstprechend einordnet.
Durch Anwendung eines Sortieralgorithmus lassen sich umgekehrt aber auch Nominalwerte in Ordinalwerte transformieren. Zum Beispiel könnten die Dokumenttitel aus Tabelle 7.3 lexikografisch sortiert werden.
Bei der Wahl einer geeigneten Visualisierung spielen die Metadaten eine wichtige Rolle. Sie geben zum Beispiel darüber Auskunft, ob man bei einer späteren Visualisierung zwischen zwei vorhandenen Werten interpolieren kann (zum Beispiel Höhenangaben) oder ob die Werte diskret sind (zum Beispiel Anzahl der betreuten Diplomarbeiten eines Professors). Neben dem Datentyp eines Wertes ist auch

die Struktur einer Tabelle ein wichtiges Metadatum. Hier ist zu beachten, dass durch Tabellen-Transformationen die zugrunde liegende Datenstruktur verändert werden kann. Die Typinformationen lassen sich in der Datentabelle ablegen, indem eine weitere Spalte hinzugenommen wird. Weitere Informationen (z.B. der Wertebereich einer Variablen) lassen sich so zusätzlich einfügen.
Die Abbildung der reinen Daten in eine Datentabelle kann sowohl mit einem Verlust als auch mit einem Gewinn an Information verbunden sein. Gerade bei Messdaten können einzelne Werte falsch sein oder fehlen. Diese gilt es vor einer Visualisierung zu korrigieren. Statistische Daten (z.B. der Mittelwert) können die Datentabelle um weitere Informationen ergänzen. Diese Werte sind in der Regel abgeleitet. Neben Werten lassen sich auch Strukturen ableiten. Für die Datentransformation bedeutet dies, dass alle vier Varianten zur Ableitung von Werten möglich sind:

- Werte → abgeleitete Werte
- Strukturen → abgeleitete Strukturen
- Werte → abgeleitete Strukturen
- Strukturen → abgeleitete Werte

Natürlich ist die Umwandlung eines Wertes in eine abgeleitete Struktur bzw. einer Struktur in einen abgeleiteten Wert wesentlich komplexer. Ein Beispiel zu den Transformationsmöglichkeiten findet sich in [Card et al. 1999, Seite 21ff].
Der schwierigste Schritt bei der Abbildung von Daten in eine visuelle Darstellung ist die Transformation der Datentabelle in eine visuelle Struktur. In der Regel sind hier viele Wege möglich, aber nur eine *ausdrucksvolle* (expressive) Abbildung ist wünschenswert, d.h. die Visualisierung gibt alle Daten – aber auch nur sie – in der Datentabelle wieder. Diese Transformation ist nicht einfach, da sich leicht unerwünschte Daten in die Visualisierung „einschleichen“ können. Die Abbildung 7.2 ist ein Beispiel einer schlechten Visualisierung.
Die Y-Achse impliziert in der Visualisierung ordinale Werte für die einzelnen Länder, die es nicht gibt. Der Variablentyp für Länder ist in der Datentabelle nominal. Man kann nicht – so wie es die Visualisierung falsch andeutet – sagen, Deutschland ist größer als Schweden.
Ein weiteres Ziel einer guten Visualisierung ist das Auffinden einer effektiven Darstellung. Die Abbildung aus 7.3 ist wenig effektiv, da man hier die zugrunde liegende Sinuskurve nicht erkennt (rot = 0.0; gelb = 0.5; orange = 1.0; violet = -0.5; blau = -1.0). Für eine effiektive Visualisierung müssen die Eigenschaften der menschlichen Wahrnehmung berücksichtigt bzw. ausgenutzt werden.

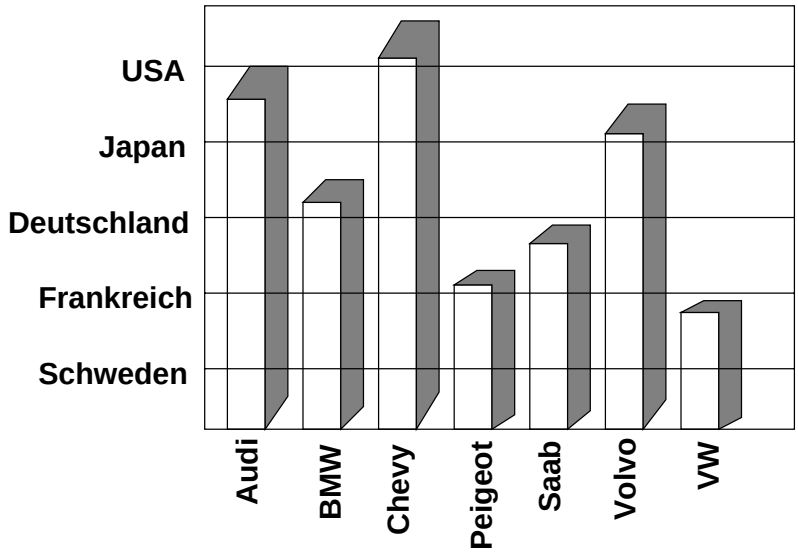


Abbildung 7.2: Eine falsche Visualisierung.

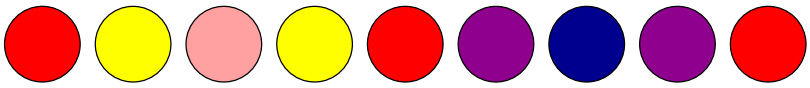


Abbildung 7.3: Die Werte einer Sinuskurve (siehe auch Farbtafel 6)

Im Rahmen der Informationsvisualisierung ist auch der letzte Abbildungsschritt von der visuellen Struktur zu einer Ansicht von Bedeutung. Verschiedene Ansichten der gleichen visuellen Struktur helfen dem Betrachter, seine Erkenntnisse zu vertiefen bzw. zu neuen Einsichten in die Daten zu gelangen.

7.2 Visualisierung mit Hilfe von Raumkonzepten

In diesem Abschnitt wird untersucht, inwiefern Raumkonzepte die Auffindung einer geeigneten visuellen Form unterstützen und ob sich auch hier ein Algorithmus finden lässt. Nach einer kurzen Einführung in die Raumkonzepte, wird ein möglicher Weg bei der Visualisierung von Daten aufgezeigt. Wegen der Raumkonzepte wird das Ergbnis dieses Weges immer eine Visualisierung in drei Dimensionen sein.

Raumkonzepte

Betrachtet man Objekte im dreidimensionalen Raum, so stehen diese in verschiedenen Beziehungen sowohl zueinander als auch zum Betrachter.

Tiefe Dreidimensionale Objekte kennzeichnen sich dadurch aus, dass sie nicht nur über die Eigenschaften Breite und Höhe, sondern auch eine Tiefe, die sich in den Raum hineinerstreckt, besitzen.

Höhe Im Raum können sich Objekte vom (zweidimensionalen) Boden lösen. Wirken die Gesetze der Schwerkraft, so müssen die Objekte über eine Gegenkraft verfügen, die sie nicht auf den Boden zurückfallen lässt. Höhe setzt auch einen Bezugspunkt voraus. Absolut gesehen kann das der Boden sein, bei relativer Betrachtung die Unterkante eines Objektes.

Lage Die Anordnung von Objekten untereinander lässt sich auch durch ihre Lage beschreiben. In der Abbildung 7.4 sind alle möglichen Lagen veranschaulicht.

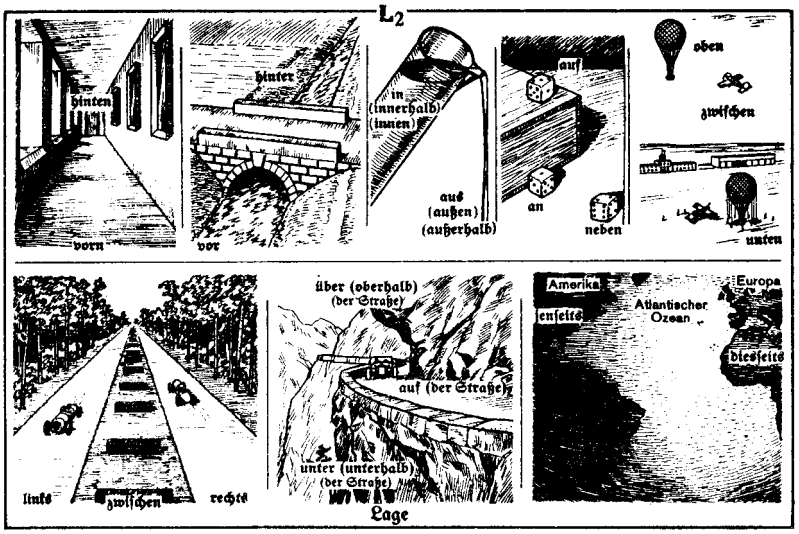


Abbildung 7.4: Lage (Quelle: [Brockhaus 1949])

Nähe Alle Objekte haben untereinander und zum Betrachter einen gewissen Abstand, der sich durch ihre Nähe auszeichnet. Ein Objekt kann nah bei einem anderen Objekt sein oder auch fern vom Betrachter. Wenn der Betrachter als einziger Teilnehmer einer 3D-Szene seine Position verändern kann, dann

ist der Abstand, also ihre Nähe, zwischen den Objekten konstant. Die Nähe zum Betrachter ändert sich jedoch mit jedem Positionswechsel.

Nähe setzt voraus, dass Objekte unterschiedlich weit voneinander entfernt sind. Nur bei unterschiedlich entfernten Objekten kann man davon sprechen, dass einige Objekte nahe sind und andere nicht. Diese Eigenschaft läßt sich zum Beispiel bei der Qualität eines Suchergebnis ausnutzen: Im Zentrum steht der Suchbegriff oder die Suchanfrage. Um dieses Zentrum werden die Ergebnisdokumente platziert. Je näher ein Dokument zum Zentrum steht, desto höher seine Trefferwahrscheinlichkeit.

Nachbarschaft Der Begriff der Nachbarschaft ist ähnlich dem der Nähe, beruht jedoch auf einem anderen Konzept. Objekte bilden Nachbarschaften dadurch, dass sie in einem gemeinsamen Zusammenhang stehen. Dieser Zusammenhang (oder Eigenschaft) unterscheidet sie von anderen Nachbarn.

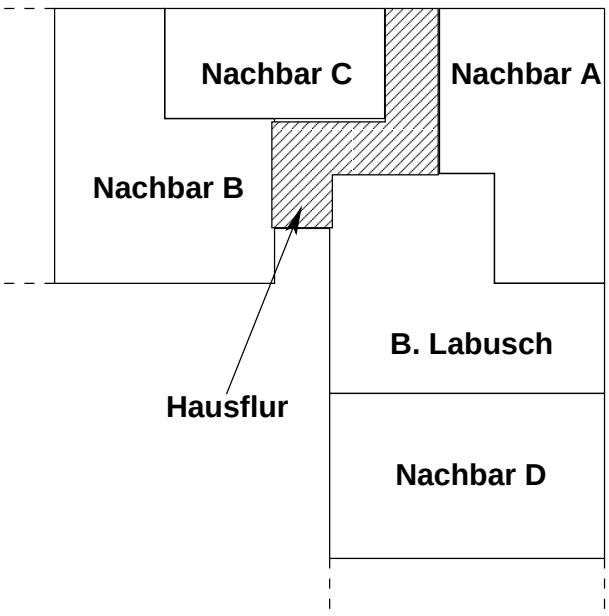


Abbildung 7.5: Nachbarschaft

Beispiel: Die Abbildung 7.5 zeigt den Grundriss meiner Wohnung. Meine Nachbarn sind A, B und C, da unsere Wohnungstüren zum gleichen Hausflur führen. Bewohner D liegt zwar näher zu meiner Wohnung als B oder C, jedoch hat D einen anderen Hausflur (sogar eine andere Hausnummer). Der gemeinsame Hausflur ist hier die Eigenschaft, die uns zu Nachbarn macht.

Der Begriff der Nachbarschaft lässt sich aus den Konzepten „Verbindung“ und „Nähe“ ableiten und halbformal als $Nachbarschaft = Verbindungen + Nähe$ formulieren.

Orte Ausgezeichnete Punkte im Raum können wir als Orte bezeichnen. An diesen Orten sind besondere Situationen anzutreffen. In der realen Welt sind dies oft geografische Besonderheiten (die sich zum Teil im Ortsnamen widerspiegeln). So ist z.B. der Ort Frankfurt an einer Furt (einer Flussüberquerung) durch den Main entstanden.

Im virtuellen Raum sind zunächst alle Punkte gleichberechtigt. Füllen wir diesen Raum mit Objekten, so ist davon auszugehen, dass einige dieser Objekte eine besondere Rolle haben werden, die sie von anderen Objekten unterscheiden werden. Hier entstehen Orte.

Verbindungen Verbindungen kennen wir aus der Graphentheorie als Kanten. Sie verbinden zwei Objekte (Knoten) auf direktem Weg.

Wege Auch das Wegekonzept ist uns aus der realen Welt bekannt. Wenn wir uns von einem Ort A zu einem Ort B bewegen wollen, so müssen wir dafür einen Weg nehmen. In den seltensten Fällen ist das die direkte Verbindung (Luftlinie), da uns andere Objekte oder physikalische Beschränkungen hiervon abhalten. Diese Wege können wir hinsichtlich unterschiedlicher Kriterien optimieren. Optimierungskriterien wären zum Beispiel objektive Maße wie Länge oder Zeit, aber auch subjektive Eigenschaften wie zum Beispiel Schönheit des Weges.

Vom Datum zur Visualisierung

Im folgenden wird ein Weg beschrieben, wie Daten unter Anwendung der Raumkonzepte visualisiert werden können. Dieser Weg enthält zwei Transformationen: Eine Abbildung von Daten auf eine geeignete Metadatenstruktur und ein zweite Abbildung, die eben diese Metadatenstruktur unter Anwendung der Raumkonzepte in eine visuelle Form transformiert. In der Grafik aus Abbildung 7.6 sind die drei Stationen dieser Abbildungsvorschrift zu sehen.
Zunächst haben wir ganz links die reinen Daten. Sie sind unstrukturiert und ohne erkennbaren Zusammenhang. Wenn wir aus diesen Daten Information oder Wissen extrahieren, werden wir diese Daten in eine Metadatenstruktur transformieren und ordnen. Danach gilt es für diese Metadatenstruktur eine geeignete Visualisierung zu finden.

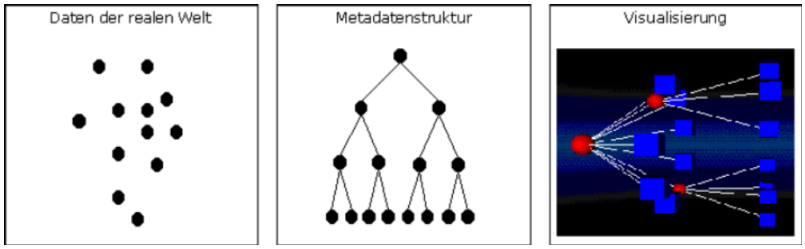


Abbildung 7.6: Vom Datum zur Visualisierung

Da bereits die Abbildung 7.6 im mittleren Schritt eine Visualisierung verwendet, lässt sich hier erkennen, dass für ein und dieselbe Metadatenstruktur unterschiedliche Visualisierungen möglich sind: Eine zweidimensionale Darstellung wie in der mittleren Abbildung oder eine dreidimensionale Darstellung wie in der rechten Abbildung zu sehen.

Schritt 1: Vom Datum zur Metadatenstruktur

In diesem Abbildungsschritt müssen zwei Vorbedingungen unterschieden werden:

- *Die Metadatenstruktur existiert vor dem Datum.* Dies trifft grundsätzlich für alle Ablagestrukturen zu.

  Beispiel 1: Bevor Daten in einer Tabelle gespeichert werden können, muss ein Datenbankschema entworfen worden sein. Dieses wird entsprechend den benötigten Anforderungen entwickelt.

  Beispiel 2: Dateien werden in allen populären Betriebssystemen (Windows, Unix, MacOS) in einer hierarchischen Verzeichnisstruktur abgelegt. Die Metadatenstruktur Baum steht hier also schon vorher fest.

- *Aus vorhandenen Daten und Metadatenstrukturen sollen neue Informationen gewonnen werden.* Hierbei handelt es sich um das klassische Aufgabengebiet des Data Mining oder von OLAP-Anwendungen (Online Analytical Processing). Hier wird mit neuen Fragestellungen auf bestehende Datenbestände zugegriffen. Diese Fragestellungen wurden beim Entwurf der Datenschemata nicht berücksichtigt.

  Stark vereinfacht stehen hierbei zwei Fragen im Vordergrund: Kann ich die gewünschte Information aus den vorhanden Daten gewinnen und welche Views (bzw. Cubes) müssen gebildet werden? Natürlich beschäftigen sich diese Theorien noch mit weiteren wichtigen Fragen (zum Beispiel der

Performance der Anfragen, da diese in der Regel auf nicht-indexierten Datenbeständen arbeiten), diese sind im Rahmen dieser Arbeit aber eher uninteressant.

Der Fall 1 muss hier nicht weiter betrachtet werden: Die Abbildungsvorschrift ist eine 1:1 Abbildung.
Interessanter ist die Frage, die Fall 2 aufwirft. Wie wird hier die Datenstruktur gefunden. Ich denke, dass es sich hierbei um einen kreativen oder intuitiven Prozess handelt, über den man sich einer initialen Datenstruktur nähert. Dieser Meinung ist auch C.J. Date

> [...] we must now also say that database design is still very much of an art, not science. [Date 1994, S. 269]

Date vertritt die Ansicht, dass man beim Datenbankdesign immer versuchen sollte, ein relationales Modell zu entwickeln. Erst wenn man hier ein klares konzeptionelles und logisches Modell aufgestellt hat, das unabhängig von der verwendeten Datenbank, dem Betriebssystem, der Programmiersprache und dem Anwendungsgebiet (Domäne) ist, sollte man die Transformation auf das Ziel-Datenbankmanagementsystem durchführen.
Ein wissenschaftlicher Ansatz zur Erstellung eines logischen Datenbankdesigns ist die Entity-Relationship-Modellierung. Sie hat allerdings einen Nachteil: Sie ist abhängig von der Domäne.
Der Transformationsprozess kann auch intuitiv sein, denn oftmals enthalten die Daten bereits einen Hinweis auf ihre Datenstruktur. So gibt es zum Beispiel in der Umgangssprache Begriffe wie

- Stammbaum
- Einkaufsliste
- Verkehrsnetz

die bereits in ihrem Namen die Datenstruktur enthalten.

Schritt 2: Von der Metadatenstruktur zur Visualisierung

Für die Visualisierung von Informationen gibt es ein eigenes Forschungsgebiet: Das der *Information Visualization*. Dieses Teilgebiet des Grafikdesign beschäftigt sich mit der effektiven Darstellung von Informationen und Daten, die meist sehr komplex und vielfältig sind (z.B. wissenschaftliche Daten, Inhalt von Datenbanken). Hier werden neue Visualisierungsmethoden entwickelt und untersucht. Man

Metainformation	Raumkonzept
Typ	Form, Gestalt
Grösse	Ausmasse (Breite, Höhe, Tiefe)
Position	Ort
Dimension	*entfällt*

Tabelle 7.4: Abbildung von Datenstruktur-Metainformationen auf Raumkonzepte

macht sich dabei Prinzipien des Layouts, der Nähe, der Anordnung und gemeinsamer visueller Eigenschaften (z.B. Farbe) zu nutze (siehe [Usability First 2001]). Ein komplettes Wissensgebiet kann natürlich nicht in einer Diplomarbeit wie dieser wiedergegeben werden. Diese Arbeit vertritt daher die Hypothese, dass sich die Metainformationen der Datenstrukturen auf Raumkonzepte abbilden lassen und so im Raum eine intuitive Darstellung des Wissens aus Datenbasen entsteht, durch die sich leicht navigieren lässt. Ob die Hypothese dem Anspruch einer einfachen und inituitiven Bedienung gerecht wird, wurde durch Usability-Tests untersucht. Die Ergebnisse dieser Untersuchungen sind in Kapitel 9 festgehalten. Bevor wir uns über mögliche Abbildungen unterhalten können, sollten wir die Raumkonzepte erläutern.

Abbildung von Metainformationen auf Raumkonzepte

Die Tabelle 7.4 stellt die Metainformationen der Datenstrukturen (vgl. Abschnitt 5.2) den Raumkonzepten gegenüber.
Für die Transformation der Metainformation Dimension in den Raum gibt es genau eine Abbildung. Alle Metadatenstrukturen werden bei diesem Ansatz in den dreidimensionalen Raum abgebildet. Z.B. wird eine Baumstruktur hier immer dreidimensional dargestellt, z.B. als Cone Tree.
Einige Beispiel für die Abbildung der Metainformationen bestimmter Datenstrukturen auf Raumkonzepte sind in Tabelle 7.5 zu sehen.

Anmerkungen Einige Abbildungen von Metadatenstrukturen auf Raumkonzepte müss von mir noch näher eingegangen werden, da diese nicht direkt aus der Tabelle 7.5 ersichtlich sind.

- *Feld:* Die Datenstruktur Feld hat die Eigenschaft, dass das Dimensionskonzept direkt übertragbar ist. Die Dimensionsübergänge sind auch diskret. Dimensionen jenseits von drei können wir uns jedoch schwer bis gar nicht

visuell vorstellen. Wenn wir als Beispiel ein dreidimensionales Feld nehmen, so wäre ein Quader oder Würfel eine intuitive visuelle Repräsentation der Datenstruktur. Elemente in diesem Feld liegen entweder im Inneren, am Rand oder an einer Ecke. In Abbildung 7.7 sind diese Orte farblich hervorgehoben.

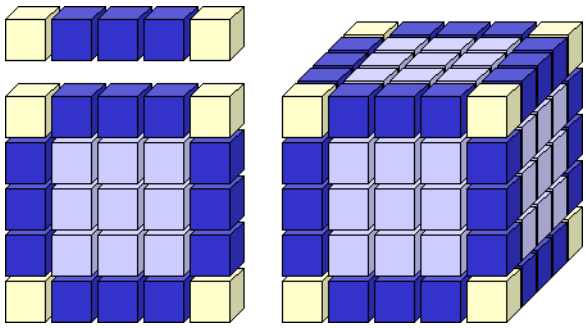


Abbildung 7.7: Besondere Orte in der Datenstruktur Feld (siehe auch Farbtafel 7)

- *Verbund:* Ein Verbund ist eine Art „Wohngemeinschaft“ für Datenstrukturen fester Größe. Raumkonzepte wie Nähe, Nachbarschaften oder Wege entfallen daher.

- *Menge:* Eine ungeordnete Menge zeichnet sich dadurch aus, dass sie keine Ordnung haben muss. Ein Element kann es in ihr auch nur einmal geben. Die einzige Gemeinsamkeit, die die Elemente haben, ist die, dass sie sich in eben dieser Menge befinden. Die Element untereinander haben keine Beziehungen. Insofern gibt es zwischen ihnen keine Verbindungen, Nachbarschaftsbeziehungen oder ähnliches.

- *Keller:* Die Datenstruktur Keller unterscheidet sich in sofern von anderen Datenstrukturen, als dass sie nur von einer äußeren Instanz manipuliert werden kann. Per Definition ist in einem Keller immer nur das oberste Element bekannt. Die anderen Elemente haben keine Kenntnis voneinander.

Raumkonzept	Datenstruktur					
	Feld	**Verbund**	**Menge**	**Liste**	**Keller**	**Baum**
Breite	Felddimension	-	-	Elementanzahl	-	Anzahl der Blätter
Tiefe	Felddimension	-	-	-	-	-
Höhe	Felddimension	-	-	-	Elementanzahl	max. Weg Wurzel -> Blatt
Lage	neben- und aufeinander	fest	nein	an- oder nebeneinander	aufeinander	auf- und nebeneinander
Nähe	isozentrisch	fest	nein	direkte Nachbarelemente	-	Verwandschaftsgrad
Nachbarschaft	direkt	Wohngemeinschaft	nein	vorheriges/nächstes Element	keine	Vater, Sohn, Geschwister
Orte	Randelemente	der Verbund selbst	die Menge an sich	Anfang, Ende, dazwischen	Kellerboden	Wurzel, Knoten, Blatt
Verbindungen	direkt	keine	keine	zwischen zwei Elementen	keine	zwischen Knoten
Wege	beliebig	keine	keine	direkt, keine Verzweigungen	keine	über Wurzel oder kürzer

Tabelle 7.5: Beispiele für die Abbildungen von Metadatenstrukturen auf Raumkonzepte

Kapitel 8

Wissensmanagement in der Praxis

In diesem Kapitel werde ich mich mit der *KnowledgeBase* von `Lufthansa Systems AS` und dem Intranet der `Fernseh AG` näher beschäftigen. Die in den Abbildungen 8.3, 8.8, 8.10, 8.13, 8.14 und 8.16 enthaltenen persönlichen Daten der Mitarbeiter mussten aus datenschutzrechtlichen Gründen unkenntlich gemacht werden. Dies ist an den verschwommenen Bereichen der Abbildungen zu erkennen.
Da die Abbildungen aus den Wissensbasen die zugrunde liegenden Strukturen zum Teil nur sehr schlecht erkennen lassen, verwende ich E/R-Diagramme, um die vorhandenen Strukturen besser hervorheben zu können.

8.1 Das Entity/Relationship-Modell

Im folgenden werde ich, bevor ich das Entity/Relationship-Modell verwende, kurz die von mir verwendeten Begriffe erklären (vergleiche auch [Date 1994]).
Das Entity/Relationship-Modell (auch E/R-Modell) besteht aus den Hauptkomponenten *Entitäten* (Entities), *Attribute* (Properties) und *Beziehungen* (Relationships). Sie sollen die in einer Datenbank enthaltenen Daten beschreiben. Abbildung 8.1 zeigt die für dieses Modell verwendeten Symbole.

Entitäten

Eine Entität ist ein unterscheidbares Objekt, wie z.B. ein Mitarbeiter oder eine Abteilung. Man unterscheidet reguläre und schwache Entitäten, wobei schwache Entitäten in ihrer Existenz von anderen Entitäten abhängig sind, d.h. die schwache Entität verschwindet, wenn die andere Entität nicht mehr vorhanden ist. Zum Beispiel kann ein Personalausweis für eine Person P nur existieren, wenn auch die

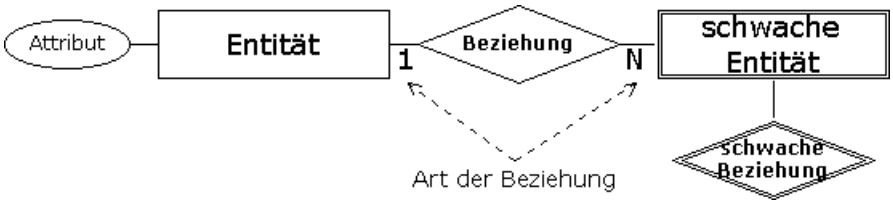


Abbildung 8.1: E/R-Symbole

Entität, also P, existiert. Eine reguläre oder starke Entität existiert unabhängig von anderen Entitäten. Jeder Entitätstyp wird als Rechteck dargestellt, wobei seine Bezeichnung in das Rechteck geschrieben wird. Bei schwachen Entitäten wird das Rechteck doppelt gezeichnet. Die Beziehung einer schwachen Entität zu der Entität, von der sie abhängt, ist immer eine m:1-Relation. In einer Datenbank werden aus den Entitäten die Tabellen gebildet.

Attribute (von Entitäten)

Entitäten haben in der Regel Attribute oder Eigenschaften.
Attribute können:

- einfach oder zusammengesetzt sein (z.B. einfach: Abteilung; zusammengesetzt: Name aus Vorname und Nachname)
- ein Schlüsselattribut sein, d.h. eindeutig. Ein Wert darf also nur einmal pro Tabellenspalte vergeben werden.
- einzelne Werte oder Mengen von Werten aufnehmen
- fehlen oder unbestimmt sein
- Basisattribute oder abgeleitete Attribute sein.

In einer Datenbank sind die Attribute die Spalten der Tabellen.

Beziehungen

Eine Beziehung (Relation) sollte verwendet werden, um eine Handlung (Aktion) zu beschreiben, an der verschiedene Entitäten beteiligt sind. Eine Relation R ist bezüglich eines in ihr vorkommenden Entity-Sets E total, falls jedes Element von E in mindestens einem Element von R vorkommt, andernfalls ist sie partiell.

Relationen

In einem E-/R-Diagramm gibt es folgende Beziehungen zwischen Entitäten:

Definition 1 (Relation vom Typ 1:M) *$R \subset X \times Y$ heisst Relation vom Typ 1:n, wenn zu jedem $y \in Y$ höchstens ein $x \in X$ gehört mit $(x, y) \in R$, aber umgekehrt zu jedem $x \in X$ mehrere $y \in Y$ gehören können.*

Formal: $\forall y \in Y, \forall x_1, x_2 \in X$ *gilt* $(x_1, y) \in R \wedge (x_2, y) \in R \Rightarrow x_1 = x_2$

Definition 2 (Relation vom Typ M:1) *$R \subset X \times Y$ heisst Relation vom Typ n:1, wenn zu jedem $x \in X$ höchstens ein $y \in Y$ gehört mit $(x, y) \in R$, aber umgekehrt zu jedem $y \in Y$ mehrere $x \in X$ gehören können.*

Formal: $\forall x \in X, \forall y_1, y_2 \in Y$ *gilt* $(x, y_1) \in R \wedge (x, y_2) \in R \Rightarrow y_1 = y_2$

Definition 3 (Relation vom Typ N:M) *$R \subset X \times Y$ heisst Relation vom Typ n:m, wenn zu einem $x \in X$ mehrere $y \in Y$ mit $(x, y) \in R$ gehören können und umgekehrt zu einem $y \in Y$ mehrere $x \in X$ mit $(x, y) \in R$ gehören können.*

Formal: $\exists x \in X, y \in Y : (x, y) \in R$

Definition 4 (Relation vom Typ 1:1) *$R \subset X \times Y$ heisst Relation vom Typ 1:1, wenn zu jedem $x \in X$ genau ein $y \in Y$ mit $(x, y) \in R$ gehört und zu jedem $y \in Y$ genau ein $x \in X$ mit $(x, y) \in R$ gehört.*

Formal: $\forall x_1, x_2 \in X, \forall y_1, y_2 \in Y$ *gilt* $(x_1, y_1) \in R \wedge (x_1, y_2) \in R \wedge (x_2, y_1) \in R \wedge (x_2, y_2) \in R \Rightarrow x_1 = x_2 \wedge y_1 = y_2$

8.2 Die KnowledgeBase von Lufthansa Systems AS

Die im folgenden als E/R-Diagramm dargestellten Bereiche der *KnowledgeBase* von `Lufthansa Systems AS` bilden nicht die tatsächliche Datenbank ab, sondern lediglich die Dinge, die man als Benutzer der *KnowledgeBase* sehen kann.
Wie in Abbildung 8.2 zu sehen ist, bietet die *KnowledgeBase* verschiedene Eintrittsportale: Die *Yellow Pages*, die *Knowledge Nets* und die *Cybrary*, die – meiner Meinung nach – die Grundsäulen dieses Wissensmanagement-Systems bilden und von mir im folgenden auch intensiver untersucht werden. Außerdem findet man noch das *Glossar*, die *Abonnements* und die *Hilfe*.
Das System bietet die Möglichkeit, on- oder offline zu arbeiten. Defaultmäßig wird eine Verbindung zum Server vorausgesetzt, aber durch das Anklicken des `GO OFFLINE`-Buttons kann man auf seinem Rechner mit der lokalen Version arbeiten.

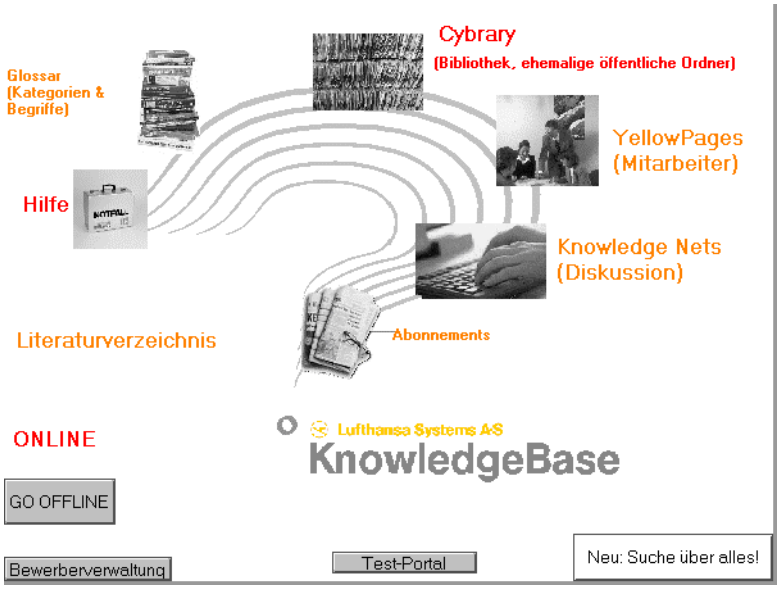


Abbildung 8.2: Startseite der *KnowledgeBase* von Lufthansa Systems AS

Die Unterschiede sind, dass man nicht mit den aktuellsten Daten arbeitet und nicht auf den Dokumentenpool zugreifen kann.
Die Architektur baut auf *Lotus Notes*[1] auf.

Yellow Pages

Die *Yellow Pages* (Abbildung 8.3) beinhalten die verschiedenen Informationen rund um die Mitarbeiter. Diese Informationen können zum Teil vom Mitarbeiter selbst angegeben werden und kommen zum anderen Teil aus einem externen System (*ProCon*). Jeder Mitarbeiter besitzt eine „elektronische Visitenkarte“, die es Vorgesetzten oder Kollegen erleichtern soll, Personen mit speziellen Fähigkeiten zu finden. Dazu trägt man im Suchfeld einfach den gewünschten Begriff ein und erhält als Suchergebnis die Mitarbeiter, auf die der Suchbegriff zutrifft.
Aus Gründen der Übersichtlichkeit habe ich die *Yellow Pages* in drei Diagramme unterteilt. Die Aufteilung erfolgte in Unternehmensdaten (Abbildung 8.4), Kontaktaufnahme (Abbildung 8.6) und Kenntnisse (Abbildung 8.5).
Bei der Kontaktaufnahme gibt es nur 1:1-Beziehungen. Zu jedem Mitarbeiter gibt es jeweils einen Datensatz zur allgemeinen Kontaktaufnahme, zur Kontakt-

[1] *Lotus Notes* ist eine Groupware-Software, die z.B. Funktionen wie Terminkalender, E-Mail, Diskussionsforen und sogar die Möglichkeit des Anlegens eigener Datenbanken bietet.

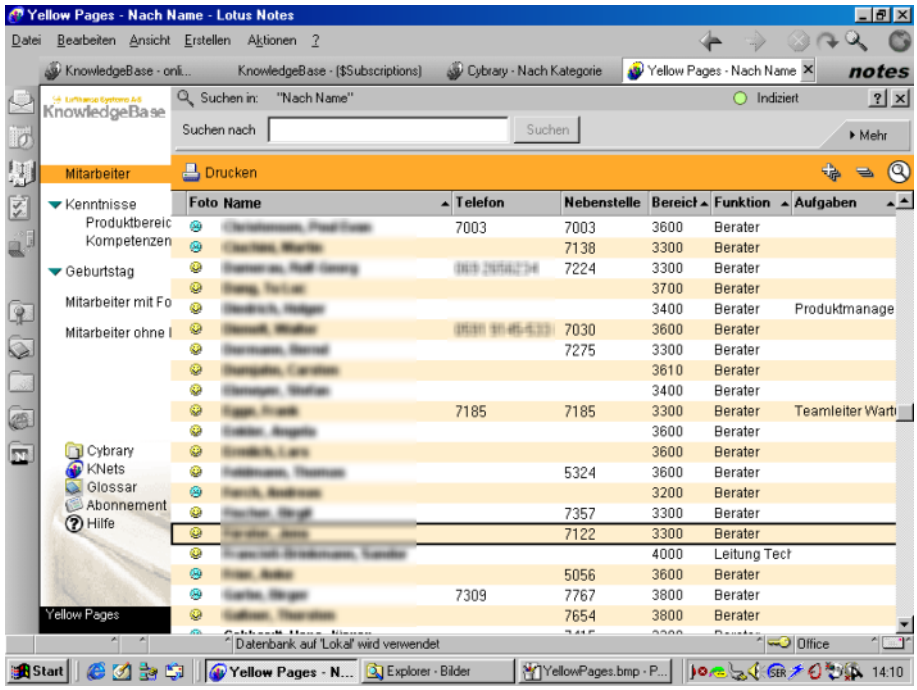


Abbildung 8.3: Die *Yellow Pages* (siehe auch Farbtafel 8)

aufnahme im Unternehmen, zur Kontaktaufnahme beim Kunden oder die private Kontaktaufnahme. Ebenso verhält es sich mit den Datensätzen, die aus dem externen System kommen: für jeden Mitarbeiter gibt es auch hier genau einen Datensatz. Bei der Kontaktaufnahme kann der Mitarbeiter auswählen, welche Form der Kontaktaufnahme ihm am liebsten ist. Zur Auswahl stehen ihm das Telefon, das Handy, E-Mail oder keine bevorzugte Kontaktaufnahme (egal).

Auch bei den Unternehmensdaten überwiegen die 1:1-Beziehungen. So wird wieder auf das *ProCon* zurückgegriffen, um für jeden Mitarbeiter einen Stammdatensatz für die *Yellow Pages* bereitzustellen. Welche Ausbildung ein Mitarbeiter genossen hat, an welchen Projekten er gearbeitet hat und welche Themen für ihn gerade aktuell sind, muss jeder einzelne Mitarbeiter aber selbst in die *Yellow Pages* eintragen. Aber genau wie es nur einen Datensatz pro Mitarbeiter aus *ProCon* gibt, so kann auch nur ein Datensatz in Bezug auf die Ausbildung jedes einzelnen Mitarbeiters angelegt werden. Die einzige 1:M-Beziehung ist die zwischen der Hauptseite der *Yellow Pages* und den Mitarbeitern, da es nur eine Hauptseite gibt, die auf viele Mitarbeiter des Unternehmens verweist.

Am interessantesten erscheinen die Beziehungen in Bezug auf das Mitarbeiter-

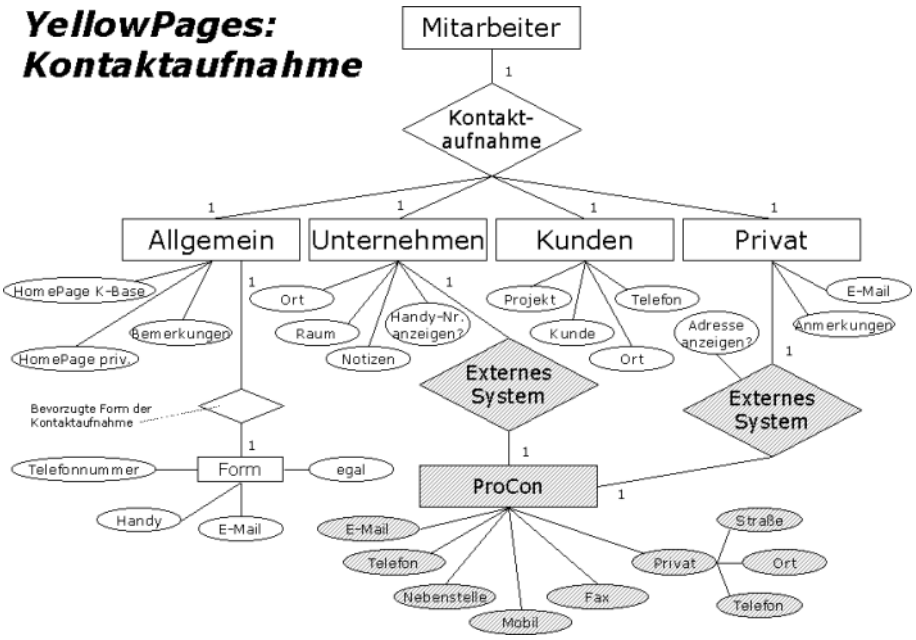


Abbildung 8.4: E/R-Diagramm der *Yellow Pages* (Teil 1)

wissen zu sein. Hier gliedert sich das Diagramm (vgl. Abbildung 8.5 und 8.7) in die großen Bereiche Schulungen und Kenntnisse. Ein Mitarbeiter kann an verschiedenen internen und externen Schulungen teilgenommen haben, wobei ihm freigestellt ist, ob diese Schulungen angezeigt werden sollen oder nicht. Die Anzahl für die externen Schulungen ist nahezu unbegrenzt, die der internen durch eine vorgegebene Liste mit auswählbaren Schulungen durch das System begrenzt. Bei den Kenntnissen wird zwischen Kompetenzen und Produktbereichen unterschieden. Die Kenntnisse werden außerdem verschiedenen Kenntnisgebieten zugeordnet. Wenn ein Mitarbeiter auf einem Kenntnisgebiet über Wissen verfügt, kann er in Form einer Werteskala angeben, wie ausgeprägt sein Wissen ist. Ob er z.B. ein Profi oder doch eher ein Anfänger auf diesem Gebiet ist. Wissen, das durch das bisherige Raster gefallen ist, wie z.B. Sprachkenntnisse, kann in Form einer 1:M-Beziehung in den *Yellow Pages* angegeben werden.

Cybrary

Die *Cybrary* (abgeleitet von Cyber und Library; siehe Abbildung 8.8) ist eine Art Bibliothek, in der Dokumente von den Mitarbeitern veröffentlicht werden können. Die Dokumente sollten im Rahmen der Arbeit entstanden sein oder anderweitig

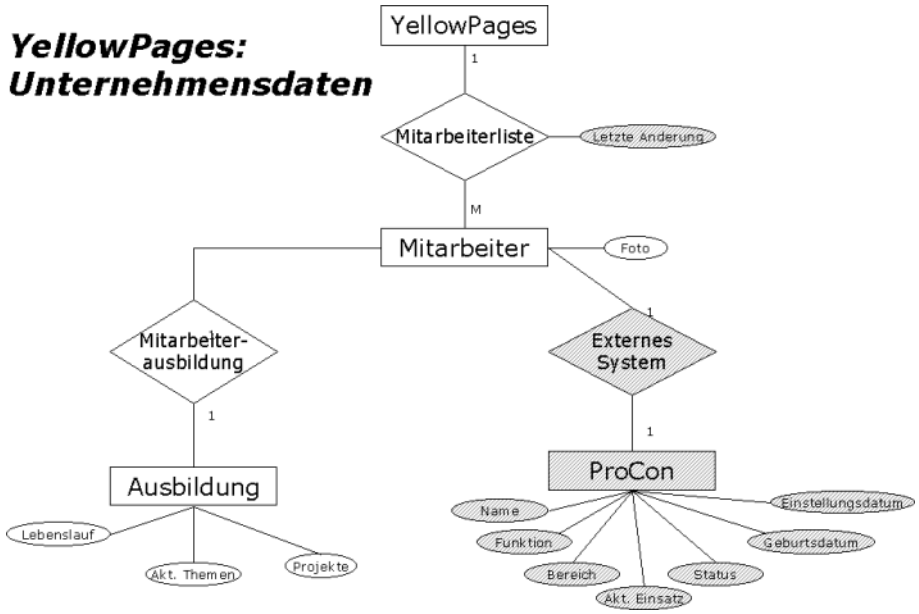


Abbildung 8.5: E/R-Diagramm der *Yellow Pages* (Teil 2)

unternehmensrelevant sein, wie z.B. Beiträge aus Fachzeitschriften.

Im Mittelpunkt der *Cybrary* stehen die Dokumente. So existiert eine 1:M-Beziehung (vgl. Abbildung 8.9) zwischen den einzelnen Mitarbeitern und den von ihnen erstellten Dokumenten, weil ein Mitarbeiter beliebig viele (das beinhaltet auch keine) Dokumente veröffentlichen kann. Eine umgekehrte, also M:1-Beziehung, besteht zwischen genau einem Dokument und beliebig vielen Mitarbeitern, wenn es um Autorenrechte, das automatische Versenden einer E-Mail-Benachrichtigung nach dem Publizieren dieses Dokuments, Leserechte oder ausgesprochene Empfehlungen von Mitarbeitern geht, die dieses Dokument als besonders lesenswert einschätzen. Durch das *Lotus-Notes-System* wird persönlich für jeden Mitarbeiter durch ein rotes Sternchen angezeigt, ob das Dokument bereits gelesen (bzw. von ihm geöffnet) wurde oder nicht. Der Übersichtlichkeit halber werden die Dokumente in Kategorien eingeordnet, die selbst wiederum in Ober- und Unterkategorien unterteilt sind. Dabei kann ein Dokument in mehreren Kategorien erscheinen, wenn es zu verschiedenen Themenbereichen passt. Ein Dokument kann vom Autor mit einem oder mehreren Dateianhängen versehen werden. Ausserdem ist es möglich, dass er ihm eine Relevanz zuordnet, also ob das Dokument z.B. besonders lesenswert ist. Diese Selbsteinschätzung kann aber auch unterbleiben. In der *KnowledgeBase* existiert ein Schlagwort-

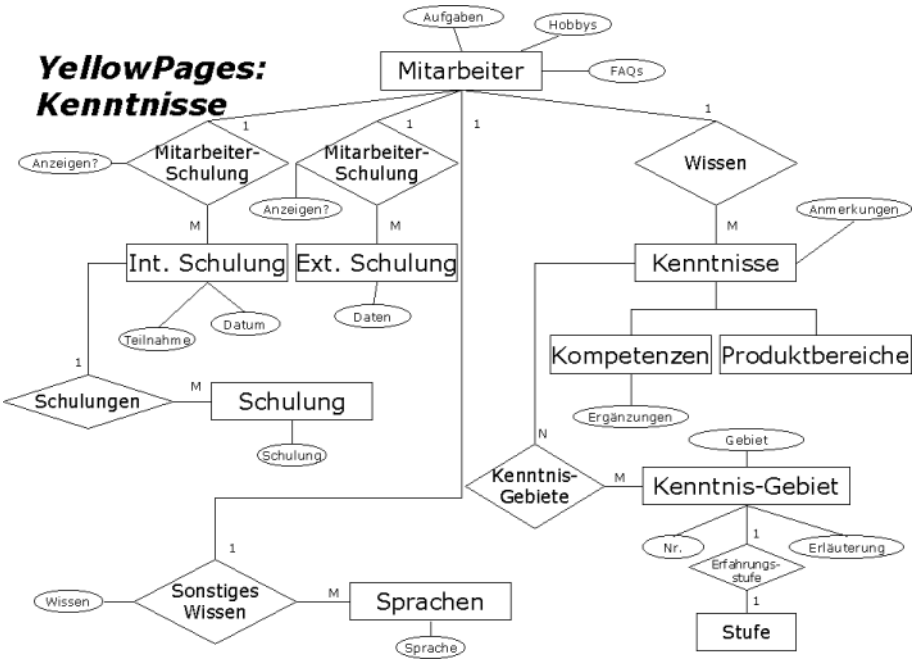


Abbildung 8.6: E/R-Diagramm der *Yellow Pages* (Teil 3)

Pool, der genutzt wird, um die Dokumente, die von den Mitarbeitern am Tage in die *Cybrary* eingebracht wurden, in einem nächtlichen Prozess zu verschlagworten. Das unterstützt den Suchprozess der Mitarbeiter nach Dokumenten immens. Die Mitarbeiter können den Prozess aber auch aktiv unterstützen, indem sie zu ihren Dokumenten explizit eine Liste von Begriffen (aus dem Schlagwort-Pool) angeben, nach denen ihr Dokument verschlagwortet werden soll bzw. bei welchen Schlagworten oder Suchbegriffen ihr Dokument nicht gefunden werden soll. Findet ein Mitarbeiter in der Auswahl der Begriffe nicht das oder die gewünschten Schlagworte, so kann er auch selbst welche der Liste hinzufügen.

Knowledge Nets

Die *Knowledge Nets* (Abbildung 8.10) erinnern stark an die aus dem Internet bekannten *Newsgroups*. Auch hier werden Diskussionen zu bestimmten Themen geführt. Primär sollen hier Themen diskutiert werden, zu denen im Internet nichts zu finden ist oder die nur unternehmensintern behandelt werden sollen.
Bei den KnowledgeNets stehen die Beiträge im Vordergrund. Jeder Beitrag ist zu einem bestimmten Thema geschrieben worden, deshalb ist jeder Beitrag un-

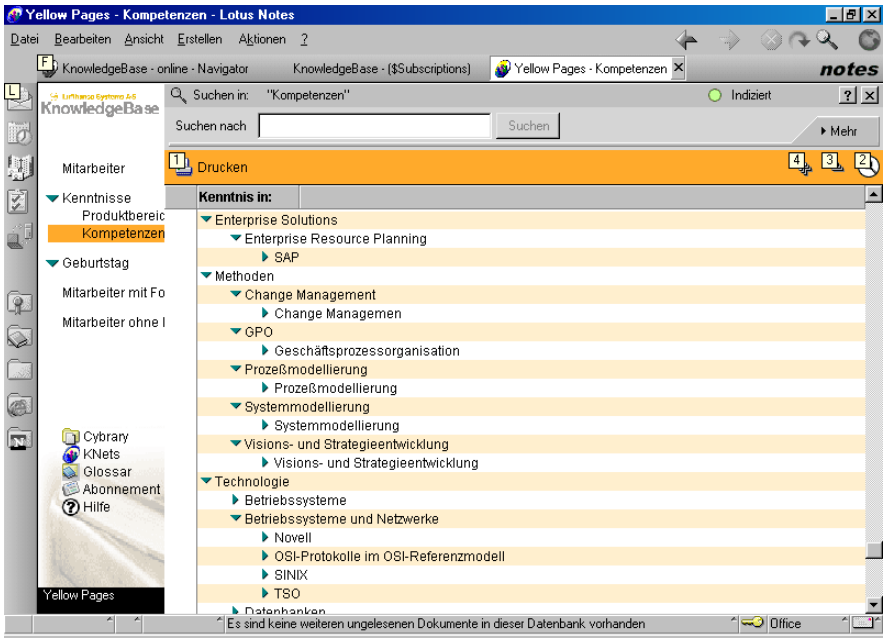


Abbildung 8.7: Die unterschiedlichen Kenntnisbereiche der Mitarbeiter

ter einer bestimmten Kategorie zu finden. Es kann beliebig viele Kategorien mit beliebig vielen Beiträgen in den KnowledgeNets geben. Jeder Beitrag kann wiederum beliebig viele Antworten enthalten (vgl. Abbildung 8.11). Genau wie in der Cybrary kann ein Mitarbeiter beliebig viele Beiträge und Antworten auf diese einbringen. Möchte ein Mitarbeiter in einer Kategorie erscheinen, so muss er mindestens ein Dokument veröffentlicht haben. Die Beiträge und auch die Antworten haben immer genau einen Inhalt, wobei der Inhalt aber eine variable Anzahl von Anhängen besitzen kann.

Welcher Bereich soll wie visualisiert werden?

Nachdem ich die *KnowledgeBase* eingehend untersucht hatte, entschied ich mich aufgrund der hohen Komplexität dafür, nur einen Teilbereich zu visualisieren. Die *Yellow Pages* schienen mir dabei ein lohnendes Beispiel abzugeben, weil ich im Verlauf meiner Arbeit die Erkenntnis gezogen hatte, dass Wissen oftmals in den „Köpfen der Mitarbeiter“ zu finden ist und über die *Yellow Pages* Mitarbeiter mit bestimmten Kenntnissen gefunden werden können. Die Abbildung 8.7 und der rechte Teil der Abbildung 8.6 zeigen genau, welcher Teil visualisiert werden

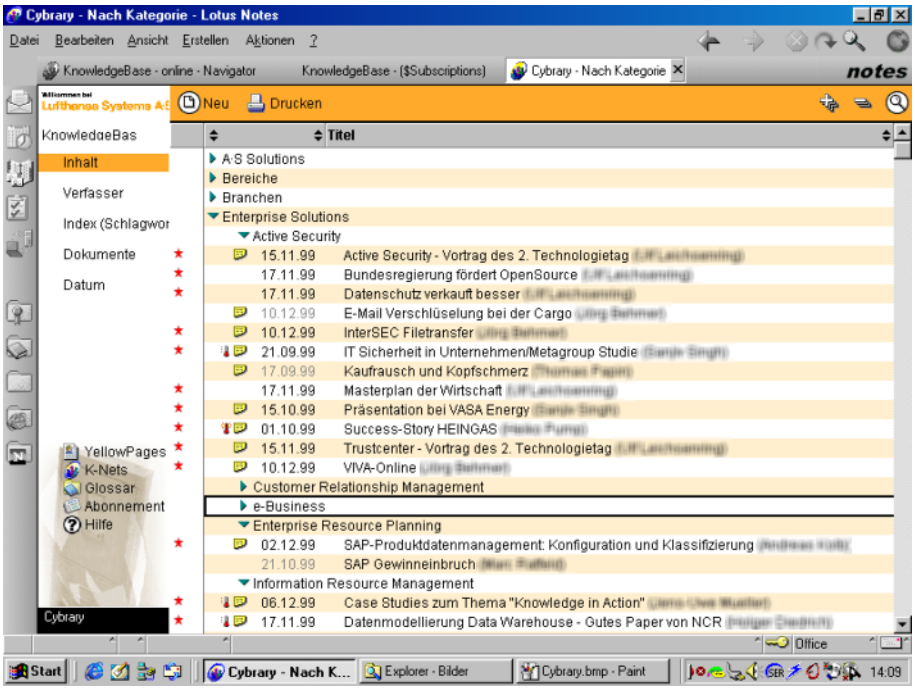


Abbildung 8.8: *Cybrary* der `Lufthansa Systems AS`

sollte.

Möchte man sich einen Überblick über die bestehenden Kenntnisgebiete des Unternehmens verschaffen, so kann die bisherige Visualisierungsform mit ihren Klappordnern schnell unübersichtlich werden, wenn alle Ordner ausgeklappt sind. In diesem Fall muss viel „gescrollt“ werden und dadurch kann der Gesamtüberblick verloren gehen.

Mein Anliegen war es, dieses Manko zu beheben. Durch das Hinzunehmen der dritten Dimension war ich in der Lage, den Raum auf dem Bildschirm besser ausnutzen zu können, um so vielleicht die Kenntnisgebiete des gesamten Unternehmens auf einer Bildschirmseite anzeigen zu können. Außerdem erschien es mir sinnvoll, das Raumkonzept mit aufzunehmen, weil die Mitarbeiter über Konzepte wie Nähe oder Nachbarschaften ähnliche Kenntnisbereiche explorieren könnten, die sie andernfalls nicht gesehen hätten.

Konkret sollten dann die ganzen Kenntnisbereiche mit Hilfe eines Cone-Trees visualisiert werden, der die hierarchische Struktur der Kenntnisbereiche gut abbilden könnte und es auch möglich machen würde, die Vielzahl dieser Bereiche lesbar auf einer Bildschirmseite präsentieren zu können.

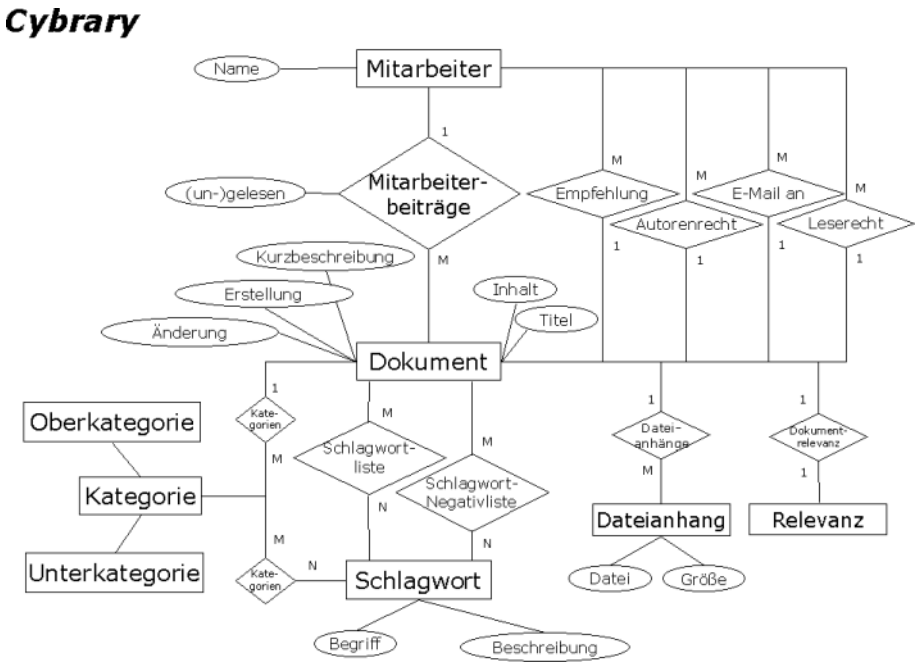


Abbildung 8.9: E/R-Diagramm der *Cybrary*

Bei den Raumkonzepten schien es mir naheliegend, eine Welt mit „richtigen“ Räumen, Türen und Fluren zu schaffen, in denen der Mitarbeiter sich bewegen könnte. In den einzelnen Räumen würden sich Bilder mit dazugehörigen Profilen der Mitarbeiter befinden, die alle Wissen über ein ganz bestimmtes Kenntnisgebiet haben. Durch das Fortbewegen in dieser Welt könnten dann andere, erst ähnliche, und dann – je weiter man sich von dem ursprünglichen Raum entfernen würde – eher unterschiedliche Kenntnisgebiete gesichtet werden.
Die Verbindung zwischen dem Cone-Tree und dem Gebäude würden die Knoten des Cone-Trees bilden, also die verschiedenen Kenntnisgebiete. Über diesen „Link“ käme man dann in den speziellen Raum mit den Mitarbeitern, die über Kenntnisse in dem Bereich verfügen, den man angeklickt hat.

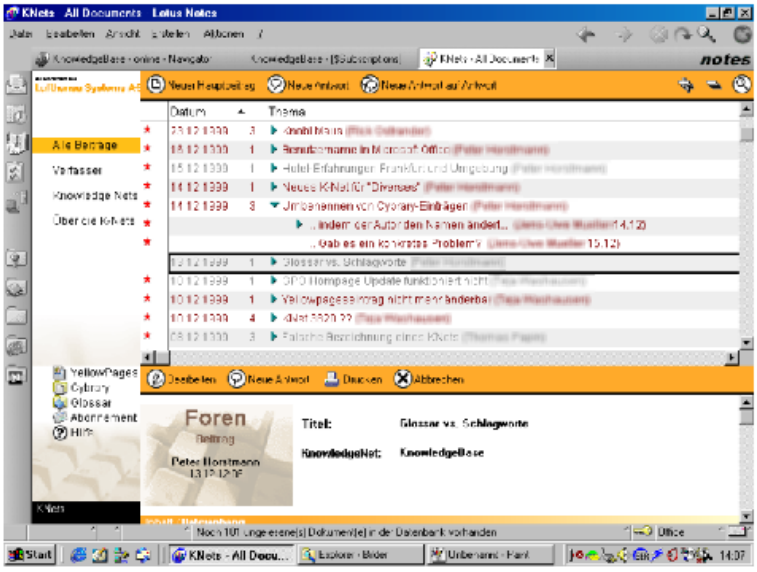


Abbildung 8.10: *Knowledge Nets* der `Lufthansa Systems AS`

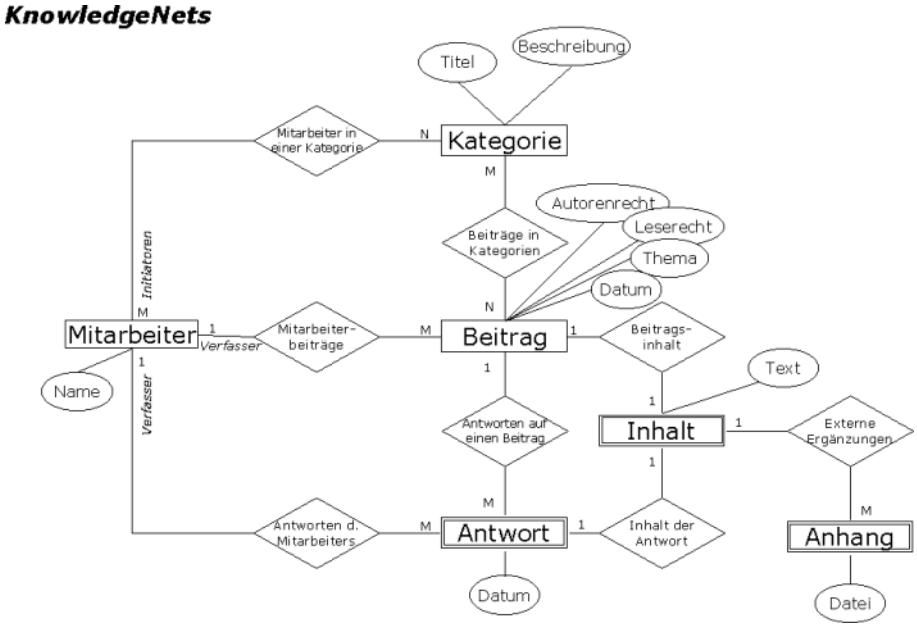


Abbildung 8.11: E/R-Diagramm der *Knowledge Nets*

8.3 Das Intranet der Fernseh AG

Das Intranet der `Fernseh AG` unterteilt sich für den Benutzer in die von jedem Mitarbeiter direkt aufrufbaren Kategorien Home, Mitarbeiterliste, Aktuelle Termine, Messen/Events, Kontakte, Eyecatcher und dpa Online (siehe auch Abbildung 8.12). Die Funktionen, die die Mitarbeiter bekleiden, sind weiter gestreut als bei der `Lufthansa Systems AS`, wo die Beratertätigkeit überwiegt. So ist es auch nicht verwunderlich, dass der Wissensaustausch zwischen den Mitarbeitern nicht über Dokumente läuft. Eine große Anzahl von Mitarbeitern nutzt das Intranet eher passiv, um z.B. Informationen über bevorstehende Drehs oder das Essensangebot der Kantine in Erfahrung zu bringen. Solche Informationen tragen einige wenige Mitarbeiter über eigens dafür programmierte Pflegeinterfaces in das Intranet ein; dazu müssen diese Mitarbeiter bestimmte Zugriffsrechte haben, die es ihnen erlauben, diese Funktionen überhaupt sehen zu können. Eine Ausnahme bilden die Kleinanzeigen, in die jeder Mitarbeiter Angebote oder Gesuche eintragen kann. Diese können dann wiederum von jedem anderen Mitarbeiter gesichtet werden.

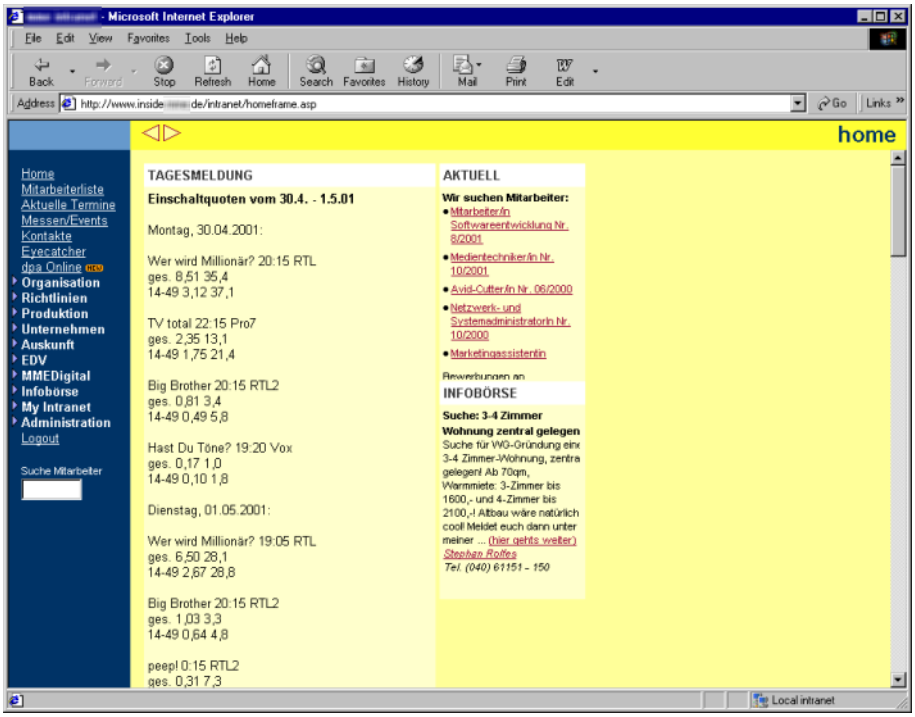


Abbildung 8.12: Die Startseite des Intranets der `Fernseh AG`

Die Mitarbeiterliste (siehe Abbildung 8.13) ist mit den *Yellow Pages* von `Lufthansa Systems AS` vergleichbar. Auch hier werden die im Unternehmen arbeitenden Personen namentlich – und alphabetisch geordnet – angezeigt (inklusive diverser Detailinformationen). Möchte man mehr über einen Mitarbeiter erfahren, klickt man auf den Namen und bekommt so auf einer nächsten Seite noch mehr Informationen angezeigt. Die genaue Art der Informationen kann Abbildung 8.14 entnommen werden.

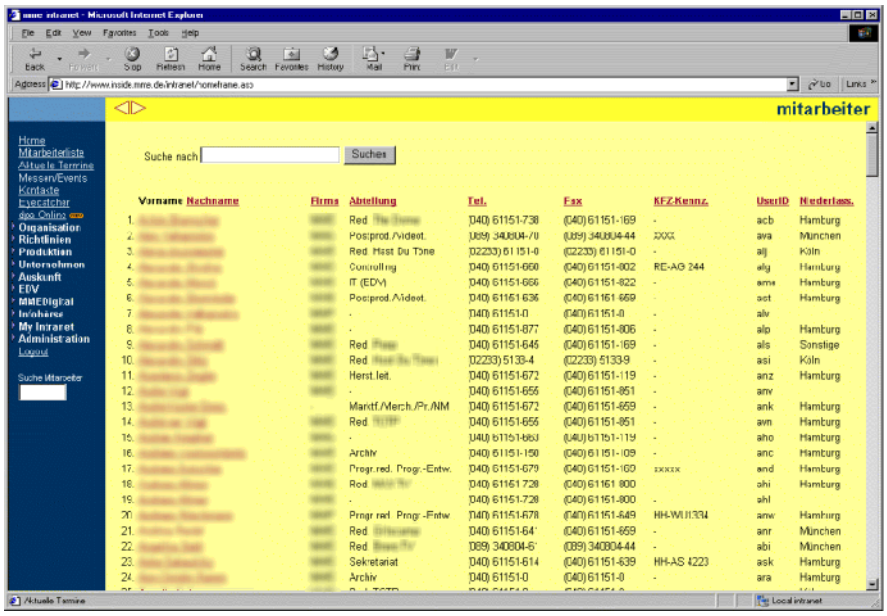


Abbildung 8.13: Mitarbeiterliste

Organigramm

Das Organigramm der `Fernseh AG` zeigt die verschiedenen Abteilungen und Redaktionen der Firma auf. Durch die hierarchische Struktur soll u.a. kenntlich gemacht werden, welcher Bereich welchem untergeordnet ist. Es ist zudem für neue Mitarbeiter gut geeignet, um sich möglichst schnell in den Firmenstrukturen zurechtzufinden zu können, denn das Unternehmen hat eine hohe Mitarbeiterfluktuation.

Abbildung 8.14: E/R-Diagramm zur Mitarbeiterliste

Welcher Bereich soll wie visualisiert werden?

Auch bei der `Fernseh AG` war ich gezwungen, nur einen Teilbereich des Intranets herausgreifen zu können, weil das gesamte Intranet zu komplex ist. Da sich aber nur ein Teilbereich auch in das Gebiet des Wissensmanagements einordnen lässt, fiel die Entscheidung leichter als bei `Lufthansa Systems AS`. Die Wahl fiel auf das Organigramm der Firma, weil ich wieder Mitarbeiterwissen kenntlich machen wollte und dies indirekt über das Organigramm abgefragt werden kann. Ein weiterer Grund war der, dass das Organigramm teilweise eine implizite und ungewollte Hierarchie ausdrückt, die von den Mitarbeitern nicht unbedingt positiv bewertet wird. So kam es immer wieder zu verhaltener Kritik darüber, warum der eigene Bereich denn unter einem bestimmten anderen liegen würde, wo doch der eigene Bereich als viel „wichtiger“ angesehen wurde. In der zweidimensionalen Anordnung hat man aber keine praktisch gangbare Möglichkeit alle

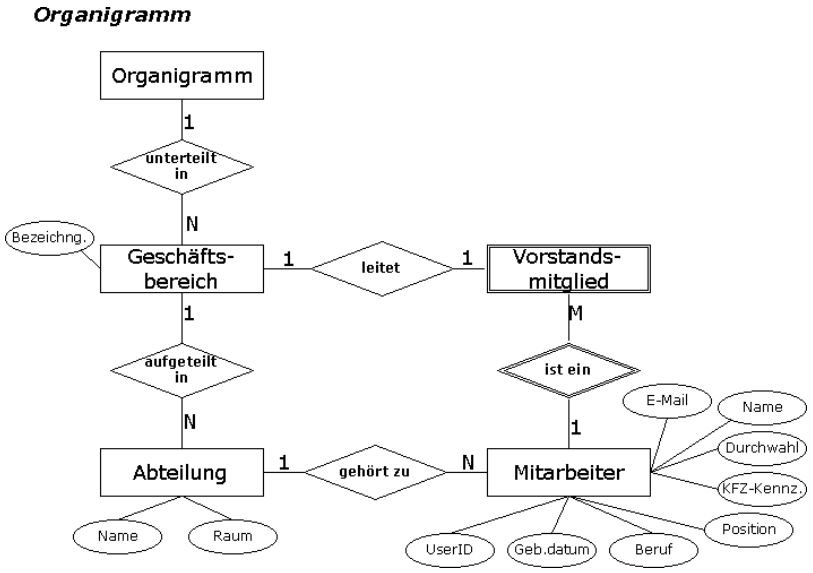


Abbildung 8.15: E/R-Diagramm des Organigramms der `Fernseh AG`.

Bereiche, die auf einer Hierarchieebene stehen, auch auf dieser anzuordnen. Es wäre lediglich denkbar, alle diese Bereiche nebeneinander anzuordnen. Praktisch ist dieser Lösungsansatz aber nicht umsetzbar, weil dies dazu führen würde, dass das Organigramm so breit werden würde, dass der Überblick vollständig verloren ginge.
Auch hier würde die dritte Dimension das Problem lösen.
In der praktischen Umsetzung sollte auch hier wieder der Cone-Tree zum Einsatz kommen. Seine Knoten würden die einzelnen Abteilungen und Redaktionen widerspiegeln und könnten so angeordnet werden, dass man einen Gesamtüberblick erhält und sich außerdem niemand darüber beschweren könnte, dass sein Bereich „unter“ einem anderen Bereich stehen würde, denn alle Bereiche, die auf einer Hierarchiestufe angesiedelt sind, würden kreisförmig um einen Kegel angeordnet sein, der alle Bereiche „gleichberechtigt“ erscheinen lassen würde.
Über einen Klick auf einen Knoten des Cone-Trees sollte man – wie es auch bei `Lufthansa Systems AS` schon angedacht war – in einen Raum gelangen, der Bilder von Mitarbeitern und dazugehörige Profile enthalten würde. Der Unterschied würde sein, dass die Gruppierung von Mitarbeitern in bestimmten Räumen nicht von ihren bestimmten Kenntnissen abhängig wäre, sondern es sich um eine Abteilungs- oder Redaktionszugehörigkeit handeln würde. Auch die Gebäudemetapher sollte sich in diesem Modell wiederfinden.

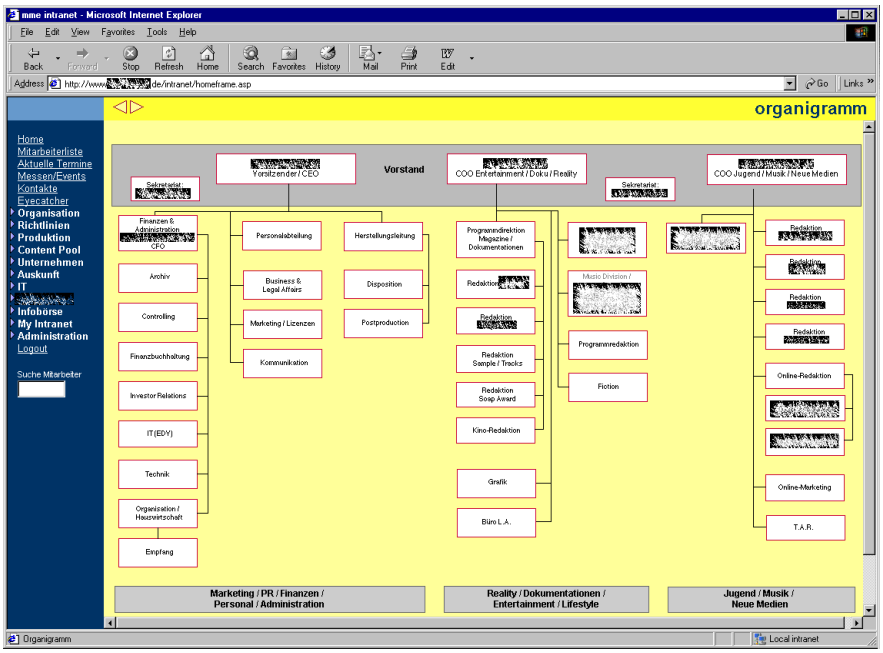


Abbildung 8.16: Das Organigramm der `Fernseh AG`. (siehe auch Farbtafel 9)

Kapitel 9

Benutzungstests

Da es aus zeitlichen Gründen nicht möglich gewesen wäre, sowohl den Prototypen[1] des *Organigramms* als auch den Prototypen der *Yellow Pages* einem Benutzungstest zu unterziehen, war ich gezwungen, eine Wahl zu treffen. Sie fiel auf das Organigramm, weil sich hier die Möglichkeit bot, mit Mitarbeitern der `Fernseh AG` diese Tests durchzuführen, also mit potentiellen zukünftigen Nutzern des Systems.
Die Benutzungstests wurden in den Räumen der `Fernseh AG` durchgeführt und verliefen in zwei zeitlich voneinander getrennten Phasen: In der ersten Phase wurde den Testteilnehmern ein Mock-up (siehe unten) vorgestellt, mit dem zwei Aufgaben gelöst werden sollten, um mögliche Systemschwächen aufzudecken. In der zweiten Phase, die etwa zwei Monate nach der Ersten lag, wurden noch einmal Teilnehmer aus der ersten Testreihe gebeten, den – nun auf einem Computer implementierten – Prototypen zu testen. Auch hier galt es wieder, Aufgaben mit Hilfe des Systems zu lösen.

9.1 Benutzungstests mit dem Mock-up

Um die potentiellen späteren Benutzer möglichst frühzeitig in den Softwareentwicklungsprozess einzubeziehen, wurde – noch bevor eine einzige Zeile Programmcode geschrieben worden war – ein Benutzungstest mit Hilfe eines sog. Mock-ups durchgeführt. Es wurde von mir auch nur dieses eine Mock-up gebaut, weil ein

[1] Ein Prototyp konzentriert sich (nach [Preece 1993, Seite 104]) auf einige Aspekte eines interaktiven Systems und ignoriert andere. Er kann sich vom fertigen System in folgenden Punkten unterscheiden: Grösse, Zuverlässigkeit, Robustheit, Vollständigkeit und Konstruktionsmaterialien.

zweites, das die *Yellow Pages* abgebildet hätte, nicht hätte getestet werden können und so keine Erkenntnisse für die weitere Entwicklung geliefert hätte.

Aufbau des Mock-ups

Ein Mock-up[2] ist eine Attrappe oder ein Modell, mit dem das spätere Produkt dargestellt werden soll (bei näherem Interesse verweise ich auf [Lindgaard 1994]). In der Regel sind Mock-ups, die für die Darstellung einer Benutzungsoberfläche erstellt werden, zweidimensional. Dies ist damit zu erklären, dass auf einem Bildschirm in der Regel nur zwei Dimensionen darstellt werden. Da ich aber die Implementierung einer dreidimensionalen Welt anstrebte, schien es mir notwendig, auch mein Mock-up um eine Dimension zu erweitern. Mit den üblichen Materialien, die für den Test einer Benutzungsschnittstelle verwendet werden, also Pappe und Papier, konnte mein Mock-up ebenfalls nur bedingt erstellt werden. Während ein zweidimensionales Mock-up von der Schwerkraft der Erde profitiert, weil es flach auf dem Tisch liegt und dann die Dialoge aufgeklappt oder ähnliche Aktionen simuliert werden können, war mir die Schwerkraft im Wege.
Im ersten Schritt musste eine dreidimensionale Konstruktion geschaffen werden, mit den Auflagen, dass den Testteilnehmern eine bildschirmähnliche Öffnung zur Verfügung gestellt werden musste und auch der Testleiter in der Lage sein musste, in das Geschehen wortwörtlich „eingreifen“ zu können. So entstand eine einem Puppentheater ähnliche Box aus Styropor, Holz, Draht, Kleber und einigen anderen Bastelmaterialien. Abbildung 9.1 zeigt diesen Aufbau.
Als nächstes sollte ein frei schwebender Cone-Tree dargestellt werden. Er wurde aus Styropor gefertigt und die einzelnen Kegel, die der Einfachheit halber als Kreise dargestellt wurden, dann mit Draht verbunden. An einem Stück umgeknickten Drahtes, der aus der Spitze des Cone-Trees herausragte, konnte er an einem der Drähte, die quer über das ganze Dach der Box verliefen, festgemacht werden. Diese vertikale Ausrichtung des Cone-Trees wurde wegen der Schwerkraft gewählt, obwohl beim Bau des Modells schon feststand, dass im fertigen System der Cone-Tree horizontal verlaufen würde, um ausreichend Platz für die Knotenbezeichner zu haben. Eine horizontale Ausrichtung war aber zu umständlich zu realisieren. Abbildung 9.2 zeigt den in der Box hängenden Cone-Tree.
Zum Schluss sollten einige Räume und ein Gang, der diese Räume miteinander verbindet, dargestellt werden. Dies sollte – neben dem Cone-Tree – den zweiten Teil des Systems darstellen. Die Konstruktion musste in die Box hineingeschoben werden können, damit es für den Benutzer den Anschein erwecken konnte, dass sich auf seinem „Bildschirm“ eine Änderung vollziehen würde. Beim Bau fanden

[2] englisch für Attrappe; to mock = nachahmen

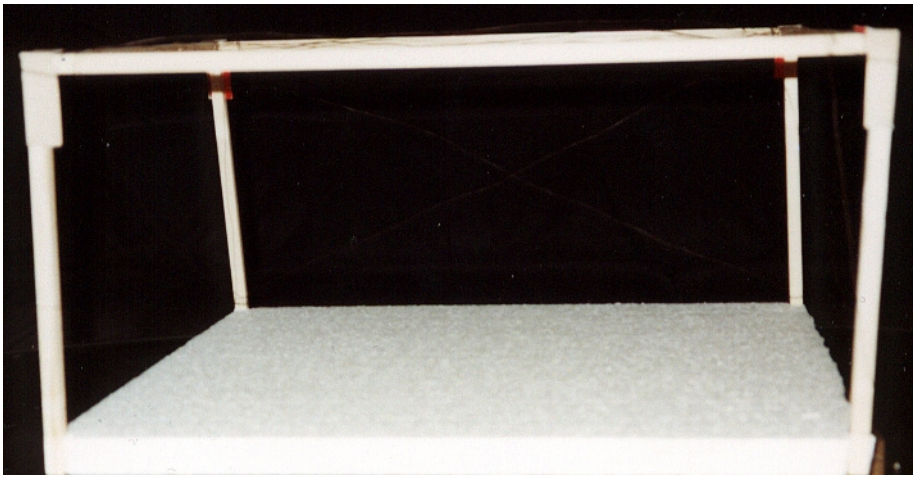


Abbildung 9.1: Die „Box“ – das Grundgerüst des Mock-ups

wieder die oben genannten Materialien Verwendung. Auf einer Styroporplatte wurden Wände aus Styropor angebracht und daran Türen aus Papier befestigt. Playmobil-Figuren, die in diesen Räumen standen, sollten die Mitarbeiter in diesen Räumen (die für die einzelnen Abteilungen standen) darstellen. Durch eingebaute Türen und Flure wurde es den Benutzern ermöglicht, sich in dem Gebäude zu bewegen, um sich z.B. einen Überblick über das Vorhandensein anderer Abteilungen verschaffen zu können. In Abbildung 9.3 kann man den Aufbau sehen. Diese Abbildung ist aus der Vogelperspektive und nicht – wie bei den letzten beiden Abbildungen – aus der Perspektive des Benutzers aufgenommen, weil man sonst nur einen geringen Ausschnitt hätte erkennen können.

Testablauf

Der Benutzungstest wurde mit insgesamt zehn Testteilnehmern durchgeführt. Alle Teilnehmer waren Mitarbeiter der `Fernseh AG` und somit zumindest ansatzweise mit einem im Intranet verfügbaren zweidimensionalen, statischen Organigramm des Unternehmens vertraut. Bei der Auswahl der Testteilnehmer konnten keine speziellen Kriterien berücksichtigt werden[3], da es ohnehin nicht ganz leicht war, Teilnehmer zu finden.

[3]Stehen ausreichend Testteilnehmer zur Verfügung, so sollte auf ein ausgeglichenes Zahlenverhältnis zwischen männlichen und weiblichen Teilnehmern geachtet werden. Außerdem sollte die Comuputererfahrung und das Fachwissen sowie das Alter berücksichtigt und in die Untersuchung mit einbezogen werden.

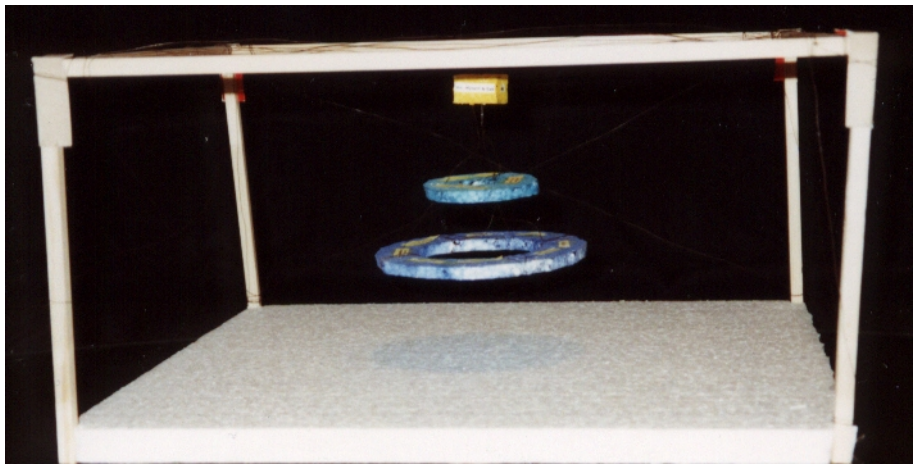


Abbildung 9.2: Der Cone-Tree (siehe auch Farbtafel 10)

Zu Beginn des Tests wurde den Testteilnehmern ein Aufgabenzettel mit folgenden zwei Aufgaben gereicht, die sie mit Hilfe des Mock-ups lösen sollten:

1. Eine Frau aus der IT-Abteilung realisiert für dich ein Intranet-Projekt. Ihr habt schon einmal darüber gesprochen, doch nun fallen dir noch ein paar wichtige Fragen ein. Leider hast du ihren Namen und auch ihre Telefonnummer vergessen. Du suchst deine Information über das neue, dreidimensionale Organigramm.

2. Du schaust dich um, welche Abteilungen es hier noch gibt.

Anschließend nahmen die Testteilnehmer vor dem Mock-up mit dem darin befindlichen Cone-Tree Platz und mussten dem Testleiter sagen, welche Aktionen er in ihrem Namen ausführen sollte. Ein kleiner Zeigestock, der den Mauszeiger simulieren sollte, diente als weiteres Hilfsmittel. Der Tester konnte den Cone-Tree beliebig rotieren lassen und mit seinem „Mauszeiger“ auf eine Abteilung klicken, wodurch er in den zweiten Teil des Systems – die Räume – gelangte. Dazu musste der Cone-Tree vom Testleiter entfernt und die Räume in die Box geschoben werden. In welchem Raum sich der Testteilnehmer dann wiederfand war von seiner vorigen Auswahl der Abteilung abhängig. In jedem Raum befanden sich die Mitarbeiter der jeweiligen Abteilung, dargestellt durch Playmobil-Figuren. Eine besonders kenntlich gemachte Figur – sie trug einen Hut – stellte den Testteilnehmer dar. Mit dieser Figur war es nun möglich, in diesem Raum, auf dem Flur oder in anderen Räumen herumzulaufen.

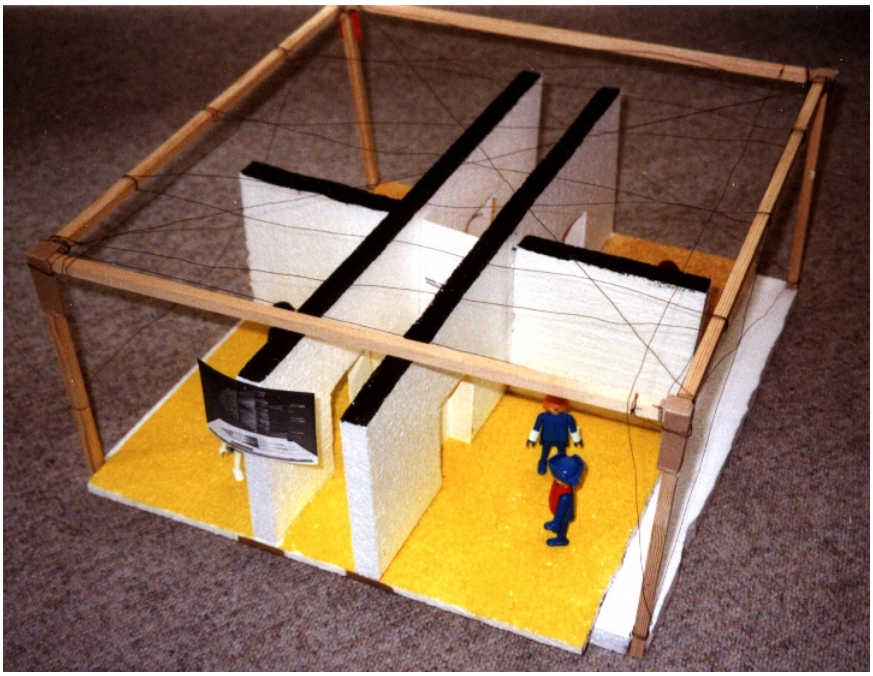


Abbildung 9.3: Ansicht der Räume aus der Vogelperspektive (siehe auch Farbtafel 11)

Nachdem ein Testteilnehmer beide Aufgaben gelöst hatte, wurde er nach seinen subjektiven Eindrücken befragt. Als Grundlage dienten hierbei die Fragen aus [Nielsen 1993, Seite 35], die von mir um die letzten beiden Fragen ergänzt wurden: Bitte bewerte die folgenden Fragen mit Werten auf einer Skala von 1 bis 5, wobei 1 für „stimme ich voll zu“ und 5 für „stimmt überhaupt nicht“ steht:

1. Die Benutzung des Systems war sehr leicht zu erlernen.
2. Die Benutzung des Systems war eine frustrierende Erfahrung.
3. Ich habe das Gefühl, dass das System eine hohe Produktivität verspricht.
4. Ich befürchte, dass viele meiner Aktionen mit dem System falsch waren.
5. Dieses System kann alle Dinge, die ich gebrauchen könnte.
6. Die Benutzung des Systems war sehr angenehm für mich.
7. Ich kann mir gut vorstellen, wie das fertige System aussehen wird.

8. Ich kann mir gut vorstellen, wie das fertige System funktionieren wird.

Ergebnisse

Die Auswertung der Antworten auf die Fragen ist der Tabelle 9.1 zu entnehmen. Jede Spalte unter der Überschrift Bewertungen steht für einen der zehn ausgefüllten Fragebögen. Aus diesen Bewertungen wurde ein Durchschnittswert berechnet, der sich in der darauf folgen Spalte (⌀) wiederfinden lässt. Bei einem Testteilnehmer (2. Spalte unter Bewertungen) musste ich davon ausgehen, dass es mit der Art des Fragebogens Probleme gab, denn obwohl dieser Testteilnehmer das System in einem Gespräch mit mir sehr positiv bewertete, vergab er für die Fragen 2 und 4 eine 1, also die schlechteste Wertung. Außerdem hatte er auf dem Fragebogen nicht nur die Wertung in Zahlen vergeben, sondern auch „Ja“ oder „Nein“ hinter die Fragen geschrieben. Ich vermute deshalb, dass der Testteilnehmer die Bewertungsskala nicht sorgfältig genug durchgelesen hat und von einer Bewertung wie in der Schule ausgegangen war (also eine 1 für „sehr gut“ hielt). In der letzten Spalte der Tabelle wurde die durchnittliche Bewertungszahl deshalb um die Antworten des zweiten Teilnehmers bereinigt, d.h. sie wurden nicht in die Berechnung des Durchschnitts mit einbezogen.

	Bewertungen										⌀	bereinigt
Frage 1	2	1	1	1	1	1	1	1	1	1	1,1	1,1
Frage 2	5	1	5	4	5	5	5	5	5	5	4,5	4,9
Frage 3	2	2	2	1	1	2	2	2	2	2	1,8	1,8
Frage 4	5	1	5	4	5	5	5	5	5	5	4,5	4,9
Frage 5	3	1	2	1	1	2	3	4	3	3	2,3	2,4
Frage 6	1	3	1	1	1	1	1	2	1	2	1,4	1,2
Frage 7	1	2	2	1	1	1	2	1	1	3	1,5	1,4
Frage 8	1	1	2	1	1	1	2	1	2	2	1,4	1,4

Tabelle 9.1: Auswertung der Fragebögen zum Mock-up

Während und nach den Tests äußerten die Teilnehmer die folgenden Vorschläge und Kritikpunkte:

- Jede Abteilung/Redaktion soll durch eine andere Farbgebung abgegrenzt werden (Idee dazu: Technik = kühl → blau).
- Ein Back-Button soll die Handhabung des Systems erleichtern.

- Eine Liste mit den aktuellen Projekten der Mitarbeiter soll angezeigt werden.
- Die Projektinhalte / -ordner sollen für Mitarbeiter, die zu einer Abteilung/Redaktion gehören, einsehbar sein.
- In einem privaten Cone-Tree sollten die Suchergebnisse gespeichert werden können.
- In einem immer sichtbaren kleinen Cone-Tree soll die derzeitige Position des Benutzers angezeigt werden.
- Die möglichen Kostenstellen und Kostenträger der einzelnen Mitarbeiter sollen angezeigt werden.
- Wie kommt man zurück, wenn man die Detailansicht des Mitarbeiters vor Augen hat? Soll man sich einfach umdrehen oder soll man weitergehen?
- Man lernt evtl. Mitarbeiter kennen, die man sonst nicht kennenlernen würde.
- Eine Suchfunktion soll eingebaut werden.
- Mit dem System kann man sich keine Liste mit allen zugehörigen Mitarbeitern einer Abteilung ausdrucken.
- Es ist – nur über das Bild – nicht klar erkennbar, welcher Mitarbeiter welche Position bekleidet.
- Eine sinnvolle Anwendung wird eher bei größeren Firmen gesehen, wo ein zweidimensionales Organigramm nicht auf eine Bildschirmseite passt.
- Die Beschriftung des Cone-Trees war durch den Winkel schlecht lesbar.

Von den oben genannten Vorschlägen wurden folgende in die Realisierung des VRML-Modells übernommen:

Farbliche Trennung Die farbliche Trennung wurde auf Bereichsebene eingeführt. Unterschiedliche Farben für einzelne Abteilungen schien aus zwei Gründen nicht sinnvoll: Zum einem gibt es genau einen Raum pro Abteilung, zum anderen hat das Unternehmen 39 Abteilungen, was zu einer entsprechenden Farbvielfalt geführt hätte.

Back-Button Dies wird durch den Browser zur Verfügung gestellt.

Position im Cone-Tree Diese Idee ließ sich im VRML-Prototypen aufgrund des immensen Aufwandes nicht realisieren.

Schließen der Detailansicht Aus dem Modell ging nicht hervor, wie die Detailansicht mit den Mitarbeiterdaten zu schließen ist. Eine entsprechende Funktion wurde für das VRML-Modell vorgesehen.

Cone-Tree-Beschriftung Wie eingangs erwähnt, stand bereits vorher fest, daß der VRML-Cone-Tree horizontal dargestellt werden sollte, um genau dieses Problem zu eliminieren.

Die Bemerkungen, die sich auf inhaltliche Aspekte bezogen (z.B. Kostenstellen oder Projektinhalte) wurden für den Entwurf des VRML-Modells nicht weiter betrachtet, da sie nicht in die Zielsetzungen meiner Untersuchungen hätten integriert werden können.

9.2 Die VRML-Prototypen

Die Prototypen des Organigramms und der Yellow Pages wurden mit Hilfe von VRML (Virtual Reality Modelling Language) auf dem Computer realisiert. Die Wahl fiel auf VRML, da man hier durch kurze Anweisungen und wenig Mathematik komplexe dreidimensionale Modelle erstellen kann. Bei der Erstellung des Organigramms flossen die Ergebnisse aus den Tests mit dem Mock-up in die Implementierung mit ein.

Einführung in VRML

Die Idee für eine dreidimensionale Benutzungsschnittstelle für das World Wide Web stammt von Mark Pesce. Nach seinen ersten Kontakten mit dem WWW musste er feststellen, dass das Web nicht dem Cyberspace entsprach, wie er ihn sich vorstellte. Er schreibt:

> Architektur ist Kunst; Raum und Orte sind wichtig für unsere Zivilisation, unsere Sprache, unser Dasein. Wenn wir wirklich den globalen Informationsorganismus aufbauen sollten [...], würden wir ihn reich an Sinneseindrücken machen müssen, so wie es die menschliche Erfahrung ist. [...] Aber wir haben Ohren zum Hören, Haut zum Fühlen, Raum, um uns in ihm zu bewegen. Wenn wir das in das Web übertragen würden, würde es menschlicher und mehr als wir selbst werden. [...] Ich wusste, dass wir mit diesen Dingen im Sinn das Web aus platten Texten und Bildern heraus in eine neue Sinnlichkeit bringen könnten, die flüssiger, aussagekräftiger und emotionaler sein würde. [Pesce 1997, Seite 45]

Zusammen mit Tony Parisi entwickelte er die grundlegenden Ideen und Spracheigenschaften, die zur Virtual Reality Modelling Language führten und 1994 in der VRML 1.0 Spezifikation festgehalten wurden. Implementiert wurde diese Spezifikation von `Silicon Graphics` auf der Basis von *Open Inventor*.
Diesem ersten Entwurf fehlten jedoch die einfachsten Möglichkeiten zur Durchführung von Interaktionen in einer Szene. Weitere Neuerungen, die in den Standard 2.0 [Web3D 1996] einflossen, sind Hintergründe, Nebel, Geländereliefs, Raumklang, Skriptfähigkeiten oder die Möglichkeit zur Definition eigener Knoten.
Die Spezifikation von VRML 2.0 wurde minimal verändert und wurde unter dem umgangssprachlichen Namen VRML 97 [Web3D 1997] als internationaler Standard ISO/IEC 14772 eingetragen.

Aufbau einer VRML-Welt

Eine VRML-Welt wird textuell in einer Datei beschrieben. In dieser können die vier Hauptkomponenten von VRML wiedergefunden werden:

- *VRML-Header.* Er steht in der ersten Zeile der Datei und definiert unter anderem die Version des VRML-Standards, zu der der Inhalt konform ist.

- *Prototypen.* VRML bietet an die 50 vordefinierte Knoten, die man mittels Prototypen um eigene Knoten ergänzen kann. VRML-Bibliotheken (zum Beispiel eine Bibliothek mit Möbelstücken) verwenden diese Prototypen.

- *Figuren, Interpolatoren, Sensoren und Skripte.* Eine VRML-Welt setzt sich aus Figuren zusammen, die mittels Interpolatoren animiert werden können. Sensoren können eine Animation auslösen oder beenden. Für komplexe Animationen kann man mit Programmskripten eigene Animationen steuern.

- *Routen.* Sensoren lösen Ereignisse aus, die in der Regel von Interpolatoren verarbeitet, in andere Ereignisse gewandelt und an die Figuren weitergeleitet werden. Der Weg dieser Ereignisse wird durch sog. Ereignis-Routen festgelegt.

Eine VRML-Welt wird als ein Baum aus Knoten dargestellt, dem sogenannten *scene graph.* Zum Beispiel gibt es Knoten für Figuren wie Quader, Zylinder oder Kugeln; Knoten für Interpolatoren, die eine Reihe von Vektor-Werten erzeugen, die für eine Animation genutzt werden können, oder Knoten, die als Sensoren fungieren und Ereignisse auslösen, z.B. wenn sich der Betrachter einem Objekt nähert oder der Cursor über ein Objekt bewegt wird. Ein ganz einfaches Beispiel zur Erzeugung einer VRML-Welt zeigt Abbildung 9.4. Das kurze Programm erzeugt die Szene aus Abbildung 9.5, die neben einer Kugel einen Kegel und einen Quader enthält. Die letztgenannten Figuren finden sich auch in meiner Implementierung wieder.
Die Knoten haben eine Reihe von Feldern, mit denen die Eigenschaften des Knotens parametrisiert werden. Zum Beispiel hat der Box-Knoten ein Feld namens *size*, mit dem die Ausmaße des Quaders festgelegt werden können.
Alle Figuren werden zunächst im Nullpunkt der VRML-Welt angeordnet. Mittels Translationen und Rotationen müssen die Objekte an ihren gewünschten Platz verschoben werden.

```
#VRML V2.0 utf8
Shape {
   appearance Appearance {
      material Material { diffuseColor 1.0 0.0 0.0 }
   }
   geometry Sphere { radius 0.5 }
}
Transform {
   translation -1.5 0.0 0.0
   children Shape {
         appearance Appearance {
            material Material { diffuseColor 0.0 1.0 0.0 }
         }
         geometry Box { size 1.0 1.0 1.0 }
   }
}
Transform {
   translation 1.5 0.0 0.0
   children Shape {
         appearance Appearance {
            material Material { diffuseColor 0.0 0.0 1.0 }
         }
         geometry Cone {
            height 1.0
            bottomRadius 0.5
         }
   }
}
```

Abbildung 9.4: Quelltext zur Erzeugung der VRML-Szene aus Abbildung 9.5

Realisierung des Organigramms als VRML-Prototyp

Die VRML-Anwendung besteht aus zwei Welten. Die erste Welt realisiert den Cone Tree. Die einzelnen Knoten (ein Knoten repräsentiert eine Abteilung) verweisen in die zweite Welt, ein Gebäude mit je einem Raum pro Abteilung.

Der VRML-Cone Tree

Der Cone Tree ist aus Cone-, Box- und Text-Knoten aufgebaut. In Abbildung 9.6 ist der fertige Cone Tree zu sehen, der das Organigramm der `Fernseh AG` widerspiegelt.
Die farbigen Kegel sind die Cone-Knoten in der VRML-Welt. Der Betrachter der Szene kann die Kegel rotieren lassen, indem er den Cursor über einen Kegel bewegt und dann die Auswahltaste drückt (z.B. die Maustaste, wenn das Ein-

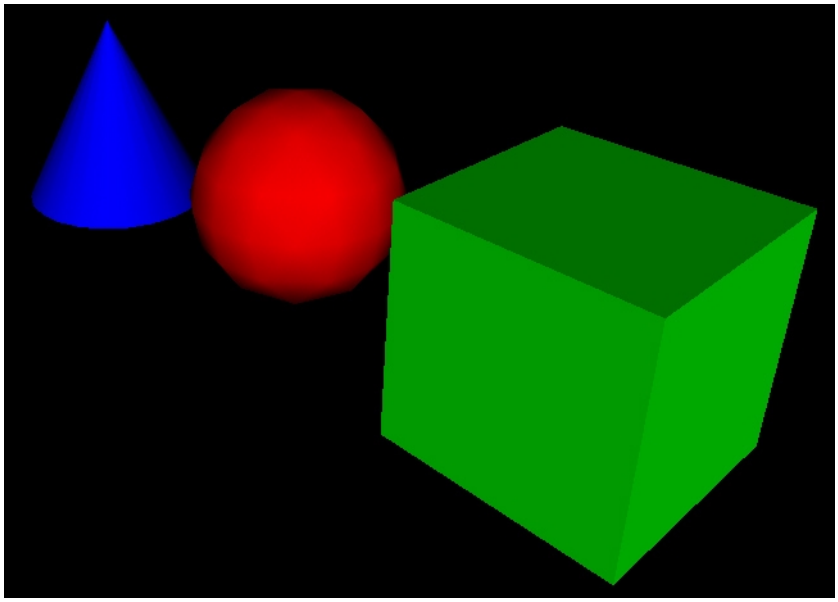


Abbildung 9.5: Die VRML-Szene zum Quelltext aus Abbildung 9.4.

gabegerät eine Maus ist). Solange die Taste gedrückt ist, rotiert der Kegel im Uhrzeigersinn. Dabei werden eventuell vorhandene Unterkegel mitgedreht. Die hierarchische Struktur findet sich analog auch in der VRML-Welt wieder und ermöglicht diesen Effekt. Technisch gesehen wird die Rotation durch einen *OrientationInterpolator*-Knoten realisiert. Die Kegel sind leicht transparent, damit im Hintergrund befindliche Objekte nicht vollständig verdeckt werden.

Die Hierarchiestufen im Organigramm der `Fernseh AG` werden durch den Einsatz von Farbe und der Kegelgröße wiedergegeben. Den drei Hauptfeldern der `Fernseh AG` wurden die drei Grundfarben Rot, Grün und Blau zugeordnet. Die Farben sollen lediglich die Zugehörigkeit einzelner Abteilungen zu den Hauptbereichen verstärken; auf Farbsehschwächen einzelner Personenkreise musste daher keine Rücksicht genommen werden (die Gruppierung ergibt sich dadurch, dass die Abteilungen direkt an einem Kegel kleben (Gesetz der Nähe)). Steigt man in der Hierarchie des Organigramms herab, so halbiert sich mit jeder Stufe der Radius des Kegels und seine Helligkeit. Die Halbierung des Kegeldurchmessers sorgt auch dafür, dass sich Unterkegel nicht überschneiden.

Ein am Kegel „angehefteter“ Bezeichner wird mittels eines Box- und eines Text-Knotens realisiert. Die Bezeichner sind sensitiv. Bewegt der Betrachter den Cursor darüber, wird der Bezeichner hervorgehoben. Ein *TouchSensor*-Knoten und

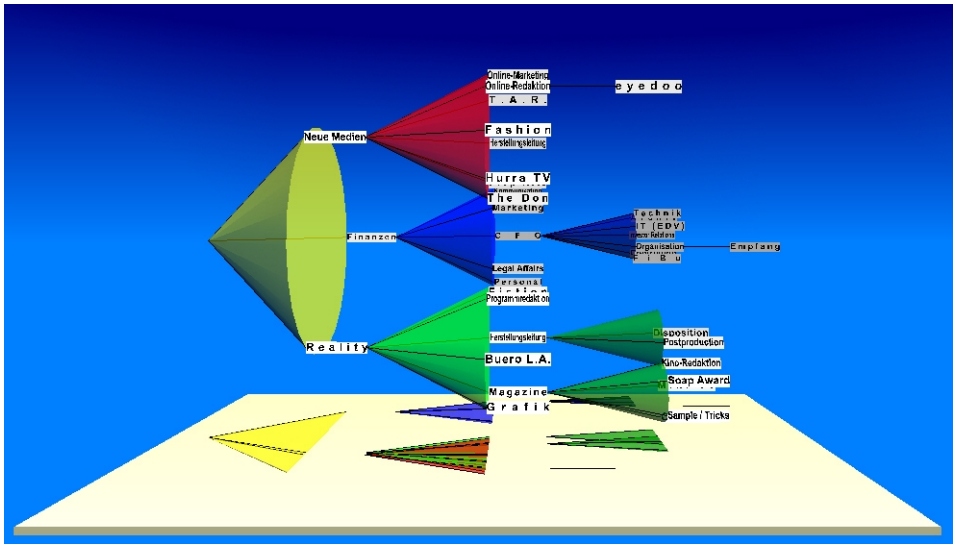


Abbildung 9.6: VRML-Cone-Tree mit den Abteilungen der `Fernseh AG` (siehe auch Farbtafel 12)

ein *Script-Knoten* mit einem kleinen Javascript-Programm setzen diesen Effekt um. Die Bezeichner sind darüber hinaus in einem *Anchor*-Knoten eingebettet, der die Beziehung zur zweiten VRML-Welt herstellt. Wählt der Betrachter einen Bezeichner aus, wird die zweite Welt mit den Räumen geladen und der Betrachter im Raum der zuvor ausgewählten Abteilung positioniert. Hier kann er sich näher über einzelne Mitarbeiter informieren oder andere Abteilungen in anderen Räumen aufsuchen.
Die Bezeichner sind in gleichmäßigen Abständen am Kegelboden „angeheftet“, wobei der erste Bezeichner immer bei Null Grad befestigt ist. Über einfache Cosinus- und Sinus-Berechnungen ergeben sich dann die Positionen der anderen Bezeichner am Rand des Kegelbodens.
Um den 3D-Effekt zu optimieren, wurde besonderer Wert auf die Beleuchtung und die Darstellung von Schatten gelegt. Schatten müssen in einer VRML-Welt eigenständig simuliert werden. Standardmäßig erzeugen die Lichter und Objekte in VRML keine Schatten, da hierfür sehr komplexe Berechnungen notwenig sind, die sofort Einfluss auf die Darstellungs- und Animationsgeschwindigkeit haben. Der Schatten des Cone Trees ist daher eine 1:1-Kopie des originalen Cone Trees, dessen Höhe jedoch auf Null reduziert und auf die Bodenfläche „gelegt“ wurde. Da es sich um eine exakte Kopie handelt, spiegeln sich die Rotationsanimationen automatisch auch im Schatten wider.

In der Cone Tree-Szene gibt es drei Lichtquellen (das *Headlight* des Betrachters wurde ausgeschaltet, um die Illumination der Szene nicht zu stören): konstantes, paralleles Licht von oben und von rechts für eine gleichmäßige Grundbeleuchtung und eine punktförmige Lichtquelle, die die Szene radial von vorne beleuchtet und deren Intensität abnimmt, je tiefer die Strahlen in den Raum gelangen. Diese Lichtquelle ist entscheidend dafür verantwortlich, dass die im Vordergrund befindlichen Bezeichner heller erscheinen als die hinteren. Dieser Effekt ist maßgeblich für die realitätsnahe Darstellung verantwortlich.
Als Hintergrundfarbe wurde ein kaltes Blau gewählt. Kalten Farben haben gegenüber anderen Fabren den Vorteil, dass sie sich weniger aufdrängen und in den Hintergrund „zurückziehen“ (siehe z.B. [Thissen 2000]).

Das VRML-Gebäude

Die zweite VRML-Welt realisiert ein Gebäude mit drei Gängen und vielen Räumen. Jeder Raum repräsentiert eine Abteilung der `Fernseh AG`. Die Gestaltung des Gebäudes und die Anordnung der Abteilungsräume stimmt nicht mit der Realität, wie sie bei der `Fernseh AG` vorzufinden ist, überein. Vielmehr erfolgt die Anordnung nach den Raumkonzepten, wie sie in Kapitel 7.2 vorgestellt wurden. Hierbei wurden die folgenden Raumprinzipien angewendet:

- *Breite, Höhe, Tiefe*: Für eine dreidimensionale Darstellung ergibt sich die Anwendung dieser Prinzipien von selbst. Die Räume wurden jedoch nicht vollständig geschlossen, sondern die Decken weggelassen, um die Navigation zu verbessern und eine Grundriss-Ansicht (siehe Abbildung 9.7) zu ermöglichen.

- *Nähe*: Die Abteilungen (bzw. ihre Räume) eines Unterzweigs liegen nebeneinander und gegenüber. Abteilungen, die sich sehr ähnlich sind, liegen in der Regel gegenüber. Verlässt man zum Beispiel die Finanzbuchhaltung, sieht man im Raum gegenüber die Controlling-Abteilung.

  Auch die Hierarchieebene spiegelt sich in der Anordnung der Räume wider. Abteilungen direkt unterhalb des Vorstands haben ihre Räume nahe dem Vorstandszimmer am Ende eines jeden Ganges.

- *Orte/Lage*: Im Gebäude gibt es vier markante Orte: Den Eingangsbereich und die Zimmer am Ende eines jeden Ganges, in denen die Vorstandsmitglieder zu finden sind. Diese bevorzugte Lage wurde gewählt, um die Hierarchieeigenschaften noch mehr zum Ausdruck zu bringen.

- *Wege*: Es gibt im Gebäude genau drei Wege, die die drei Hauptbereiche der `Fernseh AG` widerspiegeln.

Jeder Gang des Gebäudes steht für einen der Hauptbereiche der `Fernseh AG`. Die Wände des Ganges und der Räume sind dabei mit der Farbe des entsprechenden Kegels aus dem Cone Tree identisch. Der Grundriss des Gebäudes ist in Abbildung 9.7 zu sehen.

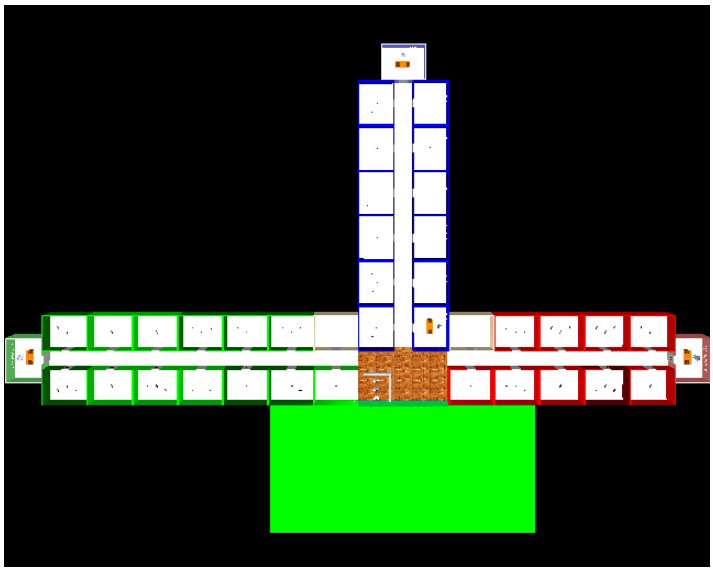


Abbildung 9.7: Grundriss des Gebäudes. (siehe auch Farbtafel 13)

In jedem Raum findet man Avatare mit den Gesichtern der Mitarbeiter aus dieser Abteilung. Diese Avatare richten sich immer zum Betrachter aus: Sie sehen ihn permanent an, um die Navigation in dem Raum zu minimieren. Die Position der Avatare im Raum spiegelt die Hierarchie in der Abteilung wieder. Der Abteilungsleiter steht direkt an der Tür, gefolgt von seinem Stellvertreter. Weiter hinten im Raum stehen die untergeordneten Angestellten (siehe Abbildung 9.8).
Wählt der Betrachter einen Avatar aus, so werden die Daten des Mitarbeiters eingeblendet (siehe Abbildung 9.9).
Für eine schnelle Naviagtion innerhalb des Gebäudes wurden eine Vielzahl von Viewpoints definiert. Über das entsprechende Menü des VRML-Viewers kann der Betrachter schnell zu diesen Standorten wechseln.
Die Rückkehr zum Cone Tree ist mittels des Cone Tree Symbols unten rechts möglich. Dieses Symbol ist immer sichtbar (es bewegt sich zusammen mit dem

Abbildung 9.8: Die IT-Abteilung der `Fernseh AG` (siehe auch Farbtafel 14)

Betrachter). Während der Usability-Tests mit den Anwendern wurde der Wunsch geäußert, dass diese verkleinerte Darstellung des Cone Trees auch die aktuelle Position im Gebäude im Cone Tree widerspiegeln sollte. Dies ließ sich jedoch nicht ohne größeren Programmieraufwand realisieren, so dass auf diese Funktionalität im VRML-Prototypen verzichtet wurde.

Realisierung der Yellow Pages als VRML-Prototyp

Die Implementierung der *Yellow Pages* wurde auf den Cone-Tree beschränkt, da es für meine Arbeit wichtiger war, einen lauffähigen Prototypen des Organigramms für die Benutzungstests zu haben. Eine vollständige Implementierung hätte – wie dies beim Organigramm der Fall war – außerdem zu aufwendigen Anonymisierungsmaßnahmen der persönlichen Mitarbeiterdaten geführt. Es wurde deshalb nur der Cone-Tree mit allen Kenntnisbereichen der Mitarbeiter implementiert, da hier keine Daten im Nachhinein aufwendig hätten verfremdet werden müssen. Zudem hätte das „Gebäude“ von der Struktur her keine erheblichen Abweichungen zu dem des Organigramms aufgewiesen und so wahrscheinlich auch keine zusätzlichen Erkenntnisse eingebracht.

Abbildung 9.10 zeigt den Cone-Tree mit den Kenntnisbereichen.

Abbildung 9.9: Profil eines Mitarbeiters.

9.3 Benutzungstests mit dem VRML-Prototypen

Testablauf

Die Benutztungstests des VRML-Prototypen wurden mit Hilfe der Evaluationsmethode des „Lauten Denkens“ durchgeführt. Zusätzlich wurden die Aussagen und Aktionen der Testteilnehmer mit zwei Videokameras festgehalten und später ausgewertet. Eine Kamera nahm dabei ständig den Bildschirm auf, mit der anderen Kamera wurde der Testteilnehmer gefilmt. Den Versuchsaufbau zeigt Abbildung 9.11.
Zusätzlich füllten die Testteilnehmer direkt nach Beendigung des Tests einen Fragebogen aus. Ich erhoffte mir dadurch, von den Benutzern eine Gesamtbewertung des Systems zu erhalten und konnte – da es sich um den gleichen Fragebogen wie bei den Benutzungstests mit dem Mock-up handelte – auch einen direkten Vergleich zwischen den Ergebnissen ziehen.
Die Aussagen der Teilnehmer wurden als Audioprotokoll festgehalten und können in Anhang A nachgelesen werden. Auf der beiligenden CD-ROM befindet sich zusätzlich das zusammengeschnittene Video von einem der Testteilnehmer. Das Video zeigt links die Testperson und rechts deren Aktionen auf dem Bildschirm. Aufgrund des hohen Aufwands, der zur Erstellung eines solchen Videos nötig ist, war es leider nicht möglich, noch weitere bzw. alle Testdurchläufe zu zeigen.
Bei den Testteilnehmern handelte es sich um eine Teilmenge der Testpersonen, die auch schon bei der Evaluierung des Mock-ups mitgewirkt hatten. Keiner der Testteilnehmer hatte zuvor Erfahrungen mit VRML oder VRML-Viewern und

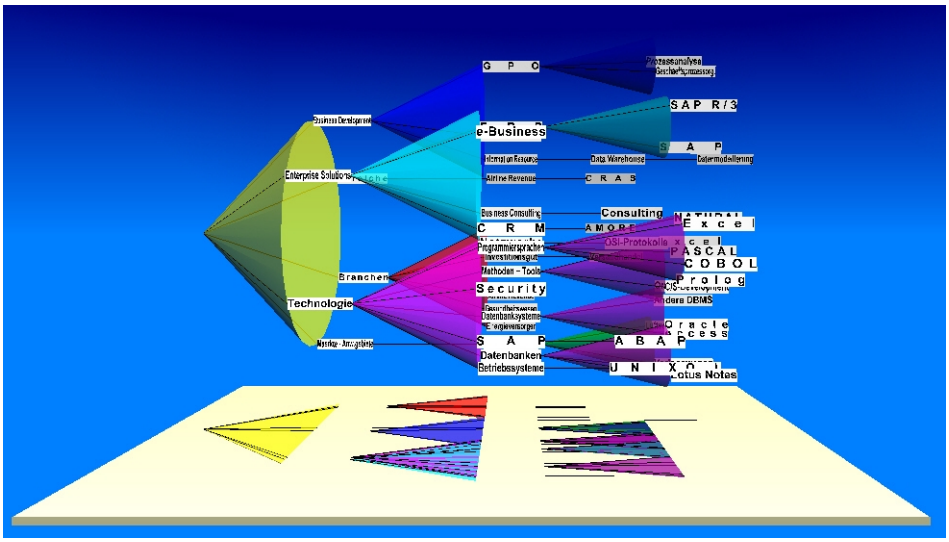


Abbildung 9.10: VRML-Cone-Tree mit den Kenntnisbereichen der `Lufthansa Systems AS` (siehe auch Farbtafel 15)

deren Handhabung gesammelt.
Für den Test wurde das VRML-Modell auf dem schnellsten Rechner der IT-Abteilung der `Fernseh AG` installiert. Wie sich beim ersten Teilnehmer zeigte, war diese Installation jedoch nicht hundertprozentig erfolgreich und verursachte deshalb im ersten Durchlauf unerwartete technische Probleme, die so auf keinem der zuvor genutzten Rechner auftraten.

Die Testteilnehmer hatten die folgenden Aufgaben zu erledigen:

1. Eine Frau aus der IT-Abteilung realisiert für dich ein Intranetprojekt. Ihr habt schon einmal darüber gesprochen, doch nun fallen dir noch ein paar wichtige Fragen ein. Leider hast du ihren Namen und auch ihre Telefonnummer vergessen. Du suchst deine Informationen über das neue, dreidimensionale Organigramm.

2. Wer ist der direkte Vorgesetzte von Irulan Shaddam[4] (Abteilung Kino-Redaktion)?

3. Welches Vorstandsmitglied ist für die Kino-Redaktion verantwortlich? Welchen Bereich leitet diese Person?

[4]Die Namen wurden aus datenschutzrechtlichen Gründen geändert.

Abbildung 9.11: Aufbau der Versuchsanordnung

4. Welcher Abteilung ist die Disposition untergeordnet?

5. Wer muss den Urlaubsantrag von Graf Hasimir Fenring[4](CFO) unterschreiben (sein direkter Vorgesetzter)?

6. Wie viele Mitarbeiter hat die Programmredaktion?

Die erste Aufgabe ist hierbei identisch mit der ersten Aufgabe aus dem Mock-up-Test. Abschließend wurden die gleichen Fragen zum subjektiven Empfinden wie auch beim Mock-up-Test gestellt.

Ergebnisse aus der Fragebögenanalyse

Nach Beendigung der Aufgaben hatten die Testteilnehmer die gleichen Fragen zu ihren subjektiven Empfinden wie beim Mock-up-Test zu beantworten (siehe Seite 159). Die Ergebnisse dieser zweiten Befragung sind der Tabelle 9.2 zu entnehmen. Die vorletzte Spalte gibt die Veränderung der durchschnittlichen Bewertungszahl im Vergleich zum Mock-up-Test wieder (vgl. Tabelle 9.1).
Die letzte Spalte zeigt das Ergebnis, wenn der zweite Fragebogen aus den Mock-up-Tests (der wahrscheinlich vrsehentlich „falsch“ ausgefüllt wurde) nicht mit in

	Bewertungen						⌀	Δ	bereinigt
Frage 1	2	3	1	1	2	2	1,8	0,7	0,7
Frage 2	5	5	3	5	5	5	4,7	0,2	-0,2
Frage 3	2	1	2	3	1	3	2,0	0,2	0,2
Frage 4	5	5	4	5	4	4	4,5	0,0	-0,4
Frage 5	3	2	2	2	4	4	2,8	0,5	0,4
Frage 6	2	2	3	1	2	3	2,2	0,8	0,9
Frage 7	1	3	2	1	2	1	1,7	0,2	0,2
Frage 8	1	2	2	1	2	1	1,5	0,1	0,1

Tabelle 9.2: Auswertung der Fragebögen zum VRML-Modell

die Auswertung aufgenommen wird. Die Bewertung des VRML-Prototypen fiel im Vergleich zum Mock-up in allen Punkten negativer aus.
Am deutlichsten sind die Abweichungen bei Frage 1 (Die Benutzung des Systems war sehr leicht zu erlernen) mit 0,7 Punkten und bei Frage 6 (Die Benutzung des Systems war sehr angenehm für mich) mit 0,9 Punkten. Inwieweit hierbei der zusätzliche Aufwand für die Erlernung der Navigation in der VRML-Welt mit dem VRML-Viewer eingeht, ließ sich im Rahmen dieser Arbeit nicht endgültig klären.

Ergebnisse aus der Auswertung der Videoaufnahmen

Die Tabelle 9.3 gibt wieder, wie lange die Testteilnehmer für die Bearbeitung der Aufgaben 1 bis 6 (A1 bis A6) benötigten. Der vergleichsweise hohe Zeitbedarf, den Thomas für die Lösung der Aufgaben 2 und 5 hatte, ist auf zwei Ursachen zurückzuführen: Bei Aufgabe 2 war aus der VRML-Welt (und auch im Original-Organigramm) nicht ersichtlich, wer der Vorgesetzte der gesuchten Person war. Diese Aufgabe wurde daher für die anderen Teilnehmer modifiziert. Bei der Bearbeitung von Aufgabe 5 stellte der Testmoderator schließlich fest, dass die Installation des VRML-Viewer Plug-Ins nicht ganz erfolgreich gewesen war. Bewegte der Testteilnehmer den Mauszeiger über die Navigationsschaltflächen des VRML-Viewers, so öffneten sich ohne Grund neue Fenster nicht oder andere, nicht nachvollziehbare Aktionen wurden ausgeführt. Aus dem Textprotokoll wird ersichtlich, dass der Testteilnehmer mehrmals den Wunsch nach einer Fortschrittsanzeige äußerte – eine direkte Folge aus der Fehlfunktion des VRML-Viewers. Der Viewer wurde auf dem Testrechner neu installiert und danach gab es keine technisch bedingten Probleme mehr.

Die Zeitangaben für Aufgabe 6 bei den Testteilnehmern Dankward und Franz sind unvollständig: eines der Videobänder war kurz vor Ende der Ausführung der Aufgaben voll. Die Werte gingen daher nicht in die Durchschnittsberechnung mit ein.
Zu Testteilnehmer Franz ist außerdem anzumerken, dass er die Ergebnisse der Aufgaben auf dem Aufgabenzettel schriftlich notierte. Dies machte sonst keine andere Testperson.
Alle Testteilnehmer fanden zu allen Aufgaben die richtigen Lösungen. Die einzige Ausnahme bildet die Aufgabe 2 bei Testteilnehmer Thomas, die vom Testmoderator aus den oben genannten Gründen abgebrochen wurde.

Testperson	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**Gesamt**
Thomas	1:22	7:52	1:56	0:37	4:21	2:00	18:08
Dankward	2:06	1:41	7:55	0:32	2:30	0:20	15:04
Michael	1:22	1:12	3:31	0:54	0:40	1:45	9:24
Carsten	1:49	1:20	2:47	1:08	0:57	2:29	10:30
Franz	2:06	2:08	2:36	0:43	0:49	1:14	9:36
Christian	1:00	0:50	1:10	0:45	0:40	1:51	6:16
Durschnitt	1:37	2:30	3:19	0:46	1:39	2:02	11:29
ohne Thomas	1:40	1:26	3:35	0:48	1:07	2:01	10:10

Tabelle 9.3: Dauer (in Minuten : Sekunden) pro Aufgabe und Durschnittswerte.

Erwähnenswert sind auch die Interaktionstechniken der Testteilnehmer mit dem System. Ein Kegel des Cone-Trees wird in Rotation gebracht, indem die linke Maustaste gedrückt und so lange gehalten wird, bis die gewünschte Rotationsposition erreicht ist. Diese Technik wurde nur von zwei Teilnehmern erkannt (siehe Tabelle 9.4). Die Mehrzahl der Teilnehmer klickte wiederholt mit der Maustaste, um die Rotation zu erreichen. Interessant war das Verhalten von Teilnehmer 1, der versuchte die Rotation durch Ziehen[5] zu realisieren. Dabei zog er die Maus immer nach unten. Das Verhalten der Anwendung war mit dieser Interaktion durchaus in Übereinstimmung zu bringen.
Im Gebäude bewegten sich nur zwei Testteilnehmer. Alle anderen veränderten die Betrachterposition maximal innerhalb der Grenzen eines Abteilungs-Raumes.
Von den Testteilnehmern wurden folgende Verbesserungsvorschläge gemacht:

- Mehr als die Hälfte der Teilnehmer wünschte sich eine Suchfunktion, weil eine gesuchte Abteilung oder Person nicht sofort gefunden werden konnte.

[5]Ziehen: Maustaste drücken, gedrückt halten und dann die Maus bewegen.

Testperson	Kegelrotation durch			Navigation im		Schließen	
	Drücken	Klicks	Ziehen	Raum	Gebäude	Text	X
Thomas	ja		ja	ja	ja	ja	ja
Dankward		ja		ja		ja	ja
Michael		ja				ja	ja
Carsten	ja	ja		ja			ja
Franz		ja				ja	
Christian		ja		ja	ja		ja

Tabelle 9.4: Unterschiedliche Systemaktionen der Testpersonen.

- Wird der Cone-Tree verlassen und dann wieder in ihn zurücknavigiert, so wird er immer wieder in seiner Ausgangsposition angezeigt. Zwei Teilnehmer hätten es als hilfreich angesehen, wenn sie die Einstellung, mit der sie den Cone-Tree verlassen hatten, auch so wieder vorgefunden hätten.

Kapitel 10

Ausblick

Die Benutzungstests mit dem VRML-Prototypen haben gezeigt, dass vor allem eine Suchfunktion gewünscht wird. Diese lässt sich mit VRML alleine jedoch nicht realisieren. Hauptgrund dafür ist die Tatsache, dass der VRML-Standard keinen Sensor-Knoten für Tastatureingaben besitzt. Als Alternative könnte man eine virtuelle Tastatur anbieten. Die Bedienung dieser virtuellen Tastatur durch Mausklicks auf einzelne Buchstaben wäre jedoch wenig effizient.
Für ein Endsystem wäre eine Datenbankanbindung wünschenswert. In den Prototypen sind sowohl die Cone-Trees hinsichtlich ihrer Anordnung und GröSSe als auch das Gebäude statisch und fest codiert im Quelltext. Ändert sich die Organisation des Unternehmens oder die Mitarbeiterzusammensetzung, so müssten derzeit die VRML-Modelle von Hand angepasst werden. Es ist zwar möglich, ein wenig Dynamik durch sog. Script-Knoten einzubinden, die JavaScript oder auch Java-Code ausführen können, jedoch können nicht ohne weiteres neue Knoten und damit neue Objekte, in eine VRML-Szene eingefügt werden. Die Objekte müssen bereits alle vorhanden sein (können aber zunächst unsichtbar bleiben).
Ein weiteres Benutzbarkeitsproblem war die Tatsache, das der Cone-Tree bei der Rückkehr aus der Gebäudeansicht immer wieder in seiner Ausgangsposition angezeigt wurde. Der Zustand lässt sich nicht speichern oder „merken“, da es sich beim Cone-Tree und beim Gebäude um zwei unabhängige VRML-Welten handelt, die in zweiverschiedenen Dateien vorhanden sind. Eine Kombination beider Welten in nur eine Datei wäre nur mit Tricks möglich (zum Beispiel die jeweils andere Welt komplett unsichtbar machen).

Der Ansatz, Metadaten auf Raumkonzepte abzubilden, um so leichter zu einer geeigneten Visualisierungsform zu kommen, ist neu. Hier sind noch viele Fragen offen, die untersucht werden können:

- Wie werden Metadaten optimal auf Raumkonzepte abgebildet?
- Lässt sich diese Transformation algorithmisch beschreiben und automatisieren?
- Enthalten Datenstrukuren noch weitere Metadaten, die nicht in Abschnitt 5.2 aufgeführt sind?
- Lässt sich eine Raumkonzept-Theorie aufstellen?

Literaturverzeichnis

[Ahlberg et al. 1994] Christopher Ahlberg und Ben Shneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In: *Proceedings CHI'94*, Seite 313–317, 479–480. Association for Computing Machinery, 1994.

[Apple 1992] Apple Computer Inc. *Macintosh Human Interface Guidelines*. Addison Wesley, New York, 1992.

[Behme et al. 1998] Henning Behme und Stefan Mintert. *XML in der Praxis: Professionelles Web-Publishing mit der Extensible Markup Language*. Addison Wesley Longman, Bonn, 1998.

[Berners-Lee 1999] Tim Berners-Lee. Challenges of the Second Decade, 1999, http://www.w3.org/Talks/1999/05/www8-tbl/slide16-0.html, Letzter Abruf: 18.02.2000.

[Bertin 1974] Jacques Bertin. *Graphische Semiologie. Diagramme, Netze, Karten*. deGruyter, Berlin, zweite Auflage, 1974.

[Brockhaus 1949] Eberhard Brockhaus. *Der Sprach-Brockhaus*. Brockhaus, Wiesbaden, 1949.

[Buzan et al. 1997] Tony Buzan und Vanda North. *Mind Mapping: Der Weg zu Ihrem persöhnlichem Erfolg*. htp, Wien, 1997.

[Card et al. 1996] Stuart K. Card, George G. Robertson und William York. The WebBook and the WebForager: An Information Workspace for the World-Wide-Web. In: *Proceedings of CHI'96*, Seite 111–117. Association for Computing Machinery, 1996.

[Card et al. 1999] Stuart K. Card, Jock D. Mackinlay und Ben Shneiderman. *Readings in Information Visualization: Using Vision To Think*. Morgan Kaufmann, San Francisco, 1999.

[Connoly et al. 1997] Dan Connoly, Khare Rohit und Rifkin Adam. The Evolution of Web Documents. In: Dan Connoly (Herausgeber), *XML: Principles, Tools and Techniques*. O'Reilly, 1997.

[Däßler et al. 1998] Rolf Däßler und Hartmut Palm. *Virtuelle Informationsräume mit VRML*. dpunkt, Heidelberg, 1998.

[Date 1994] C.J. Date. *An Introduction to Database Systems*. Addison-Wesley, Massachusetts, sechste Auflage, 1994.

[Dix et al. 1998] Alan Dix, Janet Finlay, Gregory Abowd und Russel Beale. *Human-Computer Interaction*. Prentice Hall Europe, London, zweite Auflage, 1998.

[Dublin Core 2000] Dublin Core Metadata Initiative. The Dublin Core: A Simple Content Description Model for Electronic Rsources, 2000, http://purl.org/dcyear=2000, Letzter Abruf: 18.03.2000.

[Eick et al. 1992] Stephen G. Eick, Joseph L. Steffen und Eric E. Sumner. SeeSoft – A Tool For Visualizing Line Oriented Software Statistics. *IEEE Transactions on Software Engineering*, Heft 18 (Nr. 11), Seite 957–968, 1992.

[Eppler 1997] Martin Eppler. Praktische Instrumente des Wissensmanagements – Wissenskarten: Führer durch den „Wissensdschungel". *Gablers Magazin*, Heft 8, Seite 10–13, 1997.

[Forst 1999] Annelyse Forst. Information und Wissen (Teil 1): Die neuen betrieblichen Ressourcen, 1999, http://www.doculine.com/news/1999/Februar/infowiss.htm, Letzter Abruf: 08.05.2001.

[Gould et al. 1983] John D. Gould, John Conti und Todd Hovanyecz. Composing Letters with a Simulated Listining Typewriter. *Communications of the ACM*, Heft 26/4, Seite 295–308, 1983.

[Hughes 1995] Kevin Hughes. From Webspace to Cyberspace, 1995, http://tecfa.unige.ch/guides/vrml/hughes/cspace.1.1.pdf, Letzter Abruf: 16.07.2001.

[Inxight 2001] Table Lens Walk-Through Tour, 2001, http://www.inxight.com/demos/tl_walk_thru/index.html, Letzter Abruf: 02.05.2001.

[ISO 1995] ISO. *Information Technology – Specifiacation and Standardization of Data Elements.* International Standard Organisation, 1995.

[Johnson et al. 1991] Brian Johnson und Ben Shneiderman. Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. In: *Proceedings of IEEE Information Visualization '91*, Seite 275–282. IEEE, 1991.

[Kaplan et al. 1992] Robert S. Kaplan und David P. Norton. The Balanced Scorecard – Measures That Drive Performance. *Harvard Business Review*, Heft 70 (Nr. 1), Seite 71–79, 1992.

[Keim et al. 1994] Daniel A. Keim und Hans-Peter Kriegel. VisDB:Database Exploration Using Multidimensional Visualization. *IEEE Computer Graphics and Applications*, Seite 40–49, September 1994.

[Lagemann 1987] Klaus Lagemann. *Rechnerstrukturen - Verhaltensbeschreibung und Entwurfsebenen.* Springer, Berlin, 1987.

[Lamping et al. 1996] John Lamping und Ramana Rao. The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing*, Seite 33–55, 1996.

[Lassila et al. 1999] Ora Lassila und Ralph Swick. Resource Description Framework (RDF): Model and Syntax Specification, 1999, http://www.w3.org/TR/PR-rdf-syntax/, Letzter Abruf: 18.03.2000.

[Leonhardt 1999] Claus-Peter Leonhardt. Aspekte des Wissens – Wissensmanagement, 1999, http://www.theglobalvillage.de/docs/wissensstrukt.htm, Letzter Abruf: 28.07.2001.

[Lindgaard 1994] Gitte Lindgaard. *Usability Testing and System Evaluation.* Chapman & Hall, London, 1994.

[Lockemann et al. 1987] P.C. Lockemann und J.W. Schmidt. *Datenbank-Handbuch.* Springer, Berlin, 1987.

[Mack et al. 1994] Robert L. Mack und Jacob Nielsen. *Usability Inspection Methods.* John Wiley & Sons, New York, 1994.

[Microsoft 1995] Microsoft Corporation. *The Windows Interface Guidelines for Software Design.* Microsoft Press, 1995.

[Molich et al. 1990] Rolf Molich und Jakob Nielsen. Improving a Human-Computer Dialoge. *Communications of the ACM*, Heft 33/3, Seite 338–348, 1990.

[Morse 1998] Emile Morse. Document Visualization, 1998, http://www.itl.nist.gov/iaui/vvrg/emorse/papers/soa/DocumentVisualization.htm, Letzter Abruf: 21.05.2001.

[Nielsen 1993] Jacob Nielsen. *Usability Engineering.* Academic Press, Boston, 1993.

[Nielsen 1994] Jakob Nielsen. Heuristic Evaluation. In: Jakob Nielsen und Robert Mack (Herausgeber), *Usability Inspection Methods*, Seite 25–62. John Wiley & Sons, New York, 1994.

[Nohr 2000a] Holger Nohr. Einführung in das Wissensmanagement, 2000, http://www.hbi-stuttgart.de/nohr/KM/KmAP/FHHH.pdf, Letzter Abruf: 22.08.2001.

[Nohr 2000b] Holger Nohr. Wissen und Wissensprozesse visualisieren, 2000, http://www.hbi-stuttgart.de/nohr/KM/KmAP/KnowledgeMapping.pdf, Letzter Abruf: 15.08.2001.

[Nonaka et al. 1997] Ikujiro Nonaka und Hirotaka Takeuchi. *Die Organisation des Wissens. Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen.* Campus, Frankfurt, 1997.

[Pawlowsky et al. 1997] Peter Pawlowsky und Rüdiger Reinhardt. Wissensmanagement – ein integrativer Ansatz zur Gestaltung organisationaler Lernprozesse. In: Dr. Wieselhuber & Partner (Herausgeber), *Handbuch Lernende Organisation – Unternehmens- und Mitarbeiterpotentiale erfolgreich erschließen*, Seite 145–156. Gabler, Wiesbaden, 1997.

[Pesce 1997] Mark Pesce. *VRML - Cyperspace-Welten erkunden und erschaffen.* Hanser, München, 1997.

[Preece 1993] Jenny Preece. *A Guide to Usability. Human Factors in Computing.* Addison-Wesley, Massachusetts, 1993.

[Probst et al. 1997] Gilbert Probst und Kai Romhardt. Bausteine des Wissensmanagements – ein praxisorientierter Ansatz. In: Dr. Wieselhuber & Partner (Herausgeber), *Handbuch Lernende Organisation – Unternehmens- und Mitarbeiterpotentiale erfolgreich erschließen*, Seite 129–143. Gabler, Wiesbaden, 1997.

[Probst et al. 1999] Gilbert Probst, Steffen Raub und Kai Romhardt. *Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen. 3.Auflage.* Gabler, Wiesbaden, 1999.

[Rao et al. 1994] Ramana Rao und Stuart K. Card. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context. Visualization for Tabular Information. In: *Proceedings CHI '94*, Seite 318–322. Association for Computing Machinery, 1994.

[Rekimoto et al. 1993] Jun Rekimoto und Mark Green. The Information Cube: Using Transparency in 3D Information Visualization. In: *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, Seite 125–132, 1993.

[Robertson et al. 1991] George G. Robertson, Jock D. Mackinlay und Stuart K. Card. Cone Trees: Animated 3D Visualizations of Hierarchical Informat. *Association for Computing Machinery*, Heft 34 (Nr. 2), Seite 189–194, 1991.

[Robertson et al. 1993a] George G. Robertson, Stuart K. Card und Jock D. Mackinlay. Information Visualization Using 3D Interactive Animation. *Communications of the ACM*, Heft 36 (Nr. 4), Seite 57–71, 1993.

[Robertson et al. 1993b] George G. Robertson und Jock D. Mackinlay. The Document Lens. In: *Proceedings of UIST'93*, Seite 101–108. Association for Computing Machinery, 1993.

[Robertson et al. 1994] P.K. Robertson und L.D. Ferrari. *Systematic Approaches to Visualization: Is a Reference Model Needed?* Academic Press, London, 1994.

[Romhardt 1998] Kai Romhardt. *Die Organisation aus der Wissensperspektive: Möglichkeiten und Grenzen der Intervention.* Neue betriebswirtschaftliche Forschung; Band 245, Wiesbaden, 1998.

[Roth et al. 1996] Steven Roth, Peter Lucas, Jeffrey Senn, Cristina Gomberg, Michael Burks, Philip Stroffolino, John Kolojejchick und Carolyn Dunmire. Visage: A User Interface Environment for Exploring Information. In: *Proceedings of InfoVis'96, IEEE Symposium on Information Visualization*, Seite 3–10, 1996.

[Seemann 1997] Patricia Seemann. A Prescription for Knowledge Management: What Hoffmann- LaRoche's Case can Teach Others. *Perspectives on Business Innovation*, Heft 1, Seite 26–33, 1997.

[Shneiderman 1996] Ben Shneiderman. The Eyes Have It: A Task by Data Taxonomy for Information Visualizations. In: *Proceedings 1996 IEEE, Visual Languages (Boulder, Colorado, Sept. 3-6, 1996)*, Seite 336–343. IEEE, 1996.

[Shneiderman 1998] Ben Shneiderman. *Designing the User Interface. Strategies for Effective Human-Computer Interaction.* Addison-Wesley, Massachusetts, dritte Auflage, 1998.

[Sveiby 1997] Karl-Erik Sveiby. The Intangible Assets Monitor, 1997, http://www.sveiby.com.au/IntangAss/CompanyMonitor.html, Letzter Abruf: 15.08.2001.

[Thissen 2000] Frank Thissen. *Screen Design Handbuch.* Springer, Berlin, 2000.

[Usability First 2001] Cone Tree, 2001, http://www.usabilityfirst.com/glossary/main.cgi?function=display_term&term_id=5 5.6.2001, Letzter Abruf: 26.06.2001.

[Voxel-Man 2001] Voxel-Man, 2001, http://www.uke.uni-hamburg.de/institute/imdm/idv/forschung/vm/index.en.html, Letzter Abruf: 13.07.2001.

[W3C 2000a] W3C. Metadata Activity Statement, 2000, http://www.w3.org/Metadata/Activity.html, Letzter Abruf: 17.02.2000.

[W3C 2000b] W3C. XML Schema Part 1: Structures (W3C Working Draft vom 25.02.2000), 2000, http://www.w3.org/TR/xmlschema-1/, Letzter Abruf: 26.02.2000.

[W3C 2000c] W3C. XML Schema Part 2: Data Types (W3C Working Draft vom 25.02.2000), 2000, http://www.w3.org/TR/xmlschema-2/, Letzter Abruf: 26.02.2000.

[Web3D 1996] Web 3D Consortium. VRML 2.0 Specification, 1996, http://www.vrml.org/technicalinfo/specifications/vrml2.0/, Letzter Abruf: 11.07.2001.

[Web3D 1997] Web 3D Consortium. VRML 97 Specification: ISO/IEC 14772, 1997, http://www.vrml.org/technicalinfo/specifications/vrml97/, Letzter Abruf: 11.07.2001.

[Wharton et al. 1994] Cathleen Wharton, John Rieman, Clayton Lewis und Peter Polson. The Cognitive Walkthrough Method: A Practitioner's Guide. In: Jakob Nielsen und Robert Mack (Herausgeber), *Usability Inspection Methods*, Seite 105–140. John Wiley & Sons, New York, 1994.

[Wildemann 1996] Horst Wildemann. Die Produktklinik – eine Keimzelle für Lernprozesse. *Harvard Business Manager*, Heft 1, Seite 39–49, 1996.

[Wirth 1986] Niklaus Wirth. *Algorithmen und Datenstrukturen mit Modula-2*. B.G. Teubner, Stuttgart, 1986.

[Wright 1995] William Wright. Information Animation Applications in the Capital Markets. In: *Proceedings of InfoVis, Symposium on Information Visualization, New York*, Seite 19–25. IEEE, 1995.

Anhang A

Videoprotokolle

Die Protokolle wurden mit Hilfe der Videoaufzeichnungen, die beim Testen des VRML-Prototypen entstanden, erstellt. Eine Leerzeile in einem Protokoll bedeutet, dass der Testteilnehmer länger als fünf Sekunden nichts gesagt hat. Ein Zeilenumbruch ist dort, wo nur eine kurze Pause, die weniger als fünf Sekunden lang war, entstand. Kommentare oder Einwürfe des Testleiters sind in kursiver Schrift dargestellt.
Die in den nachfolgenden Videoprotokollen genannten Namen wurden von mir aus datenschutzrechtlichen Gründen geändert.

Testteilnehmer 1: Thomas

Zwischenzeit: 0:00
Eine Frau aus der IT-Abteilung realisiert für dich ein Intranetprojekt. Ihr habt schon einmal darüber gesprochen, doch nun fallen dir noch ein paar wichtige Fragen ein. Leider hast du ihren Namen und auch ihre Telefonnummer vergessen. Du suchst deine Informationen über das neue, dreidimensionale Organigramm.
Na, dann wollen wir mal los. Dann wollen wir mal die EDV suchen. EDV. EDV. Das sieht irgendwie nicht danach aus; Medien, Reality. Finanzen.
Tja.
Da würde ich die ja nirgendwo einsortieren, oder?

Und wie es tatsächlich so sein soll. Obwohl Finanzen?

Ist ein bisschen unübersichtlich, die ganze Geschichte.

IT, EDV, da seh' ich sie doch gerade.

IT, EDV.

Zwischenzeit: 1:00

Tja, so viele Frauen scheint es hier ja nicht zu geben. Ohh! Upps! Dann habe ich vermutlich schon mal mit Birgit Labusch gesprochen. Und dann hätte ich jetzt noch die Telefonnummer.

Ja, das wäre dann Aufgabe 1.

Zwischenzeit: 1:22

Aufgabe 2. Wer ist der direkte Vorgesetzte von Irulan Shaddam, Abteilung Kinoredaktion.
Zurück.

So. Go. View.
Tja.
Restore war's, glaub ich. Aber irgendwie. Da war's.

Zwischenzeit: 2:00

OK.
Versuchen wir mal jetzt Online und Abteilung Kinoredaktion.

Neue Medien. Nicht unbedingt. Reality?

Hallo?

Da hab ich mich wohl verklickt.

Ist ein bisschen unschön, dass man gar keine Rückmeldung kriegt.

Grafik. Magazine.

Programmredaktion. Programmredaktion? Kinoredaktion, da seh ich sie zufällig im Hintergrund.

Zwischenzeit: 3:00

Klar.

Geht nicht; ...da reingucke...
Nicht wirklich, oder?

Er pfeift (siehe Abbildung A.1)

Abbildung A.1: 3:32: Thomas wartet pfeifend auf eine Reaktion des Systems. Anhand der weißen Bilder ist die Fehlkonfiguration des VRML-Viewers zu erkennen. Hier hätten die Gesichter der Mitarbeiter zu sehen sein müssen.

Auf die Gefahr hin mich zu wiederholen, das ist wirklich blöde, dass man keine Rückmeldung kriegt, dass der Kasten gerade was macht.

Zwischenzeit: 4:00

Dann gehören die hierzu.
Kinoredaktion gehört zu Magazine, dann finde ich da wohl auch den direkten Vorgesetzten.

Tja, ist das so, ist es nicht so?

Schwer zu sagen.

Zwischenzeit: 5:00

Pfeift

Zwischenzeit: 5:30

Mmm.

Reality.

Zwischenzeit: 6:00

Also ehrlich gesagt, ich bin ein bisschen überfragt, wie ich den direkten Vorgesetzten herausbekommen soll.

Pfeift

Magazine? Reality? Wer weiß es...

Leitung Programmredaktion.

Zwischenzeit: 7:00

BIRGIT: In einem Raum ist immer eine spezielle Person, wenn du dir mal die Programmredaktion anschaust.

In einem Raum? OK.

Pfeift

Zwischenzeit: 8:00

Kino-Redaktion. Wenn man so will ist dieses Bild größer als die anderen. Vielleicht soll mir das einen Hinweis geben?

Redakteure.

Sehr hübsch. Von den Redakteuren kann mit Sicherheit keiner Vorgesetzter sein.

BIRGIT: ???? (war nicht zu verstehen)

Zwischenzeit: 9:00

BIRGIT: ??? (war nicht zu verstehen)

Zwischenzeit: 9:14
OK, gut, dann mach' ich weiter mit Aufgabe 3.

Was ist hier jetzt los?

Ich sag' es einfach jetzt zum dritten Mal. Es wär mir irgendwie lieber, wenn hier so 'ne Yellow Box erscheinen würde, wo so'n Balken von links nach rechts wandert. Damit man ungefähr weiß warum, wie lange man noch warten muss.

Zwischenzeit: 10:00

Aber so bin ich irgendwie geneigt, und versuch auf den Restore-Knopf zu drücken, weil sich das Ding genauso gut aufgehängt haben kann.

Ist aber nicht der Fall. *(siehe Abbildung A.2)*

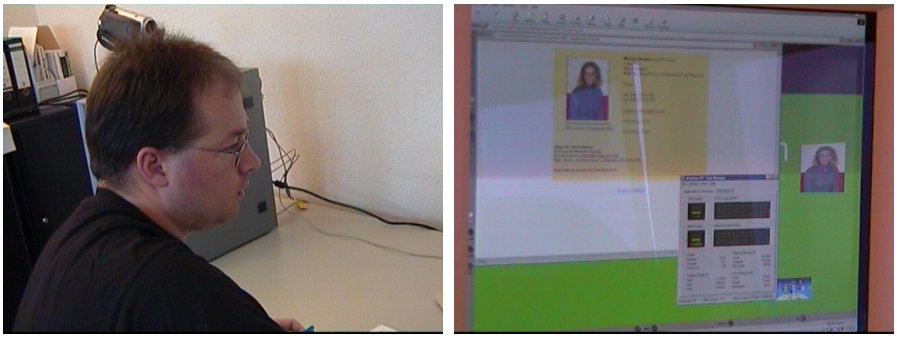


Abbildung A.2: 10:24: Thomas hat den Eindruck, die Anwendung habe sich „aufgehängt" und startet den *Taskmanager*.

Aha.

Zwischenzeit: 10:52
Jetzt sind wir wieder da. Aufgabe 3: Welches Vorstandsmitglied ist für die Kinoredaktion verantwortlich? Dürfte relativ einfach sein. Das scheint der Vorstand zu sein, der auch für Reality zuständig ist.

Zwischenzeit: 11:00

Ein Leto Atreides. Welchen Bereich leitet diese Person? Habe ich schon gesagt.

Zwischenzeit: 11:10
Aufgabe 4: Welche Abteilung, welcher Abteilung ist die Disposition untergeordnet, welcher Abteilung ist die Disposition untergeordnet. Da seh' ich sie, das ist...

Wenn ich da hinfahre ist auch immer ein bisschen störend, dass das Teil in so 'nem

Grundzustand erscheint.

Geh ich mal auf Nummer sicher. Herstellungsleitung. Herstellungsleitung und Disposition.

Ist also der Herstellungsleitung untergeordnet.

Zwischenzeit: 11:47
Aufgabe 5: Wer muss den Urlaubsantrag von Graf Hasimir Fenring unterschreiben? Ja?

Graf Hasimir Fenring? Ich glaube Finanzen, oder?
Nein, das ist cool, da bin ich also etwas überfragt, für welchen Bereich der nun zuständig ist.

Zwischenzeit: 12:00

Aber wenn er einen Vorgesetzten hat, dann kann er ja nicht ganz oben sein. Logischerweise.

Ne Suchfunktion wär' schön.

So müsste ich hier jetzt alles anklicken. Das kann dann doch ein bisschen nervig sein.

Zwischenzeit: 13:00

Graf Hasimir Fenring. Alle anderen `Fernseh AG`-Mitarbeiter werden natürlich wissen, wo er sitzt. Da hab' ich jetzt ein bisschen Pech gehabt. Hilf mir doch mal bitte.

BIRGIT: Der ist auch der CFO, der Chief Financial Officer.

Dann kann er aber keinen direkten Vorgesetzten haben. Das Ding sagt mir, das ist Stefan Eishold, der Vorsitzende für die Finanzen.

Wo mag ich dann noch den Graf Hasimir Fenring finden. Ich versuch's noch mal.

BIRGIT: ??? (war nicht zu verstehen)

Zwischenzeit: 14:00

??? *(war nicht zu verstehen)*

BIRGIT: ??? (war nicht zu verstehen)

Hmm.

Ach, natürlich.

Ja, alles klar. Rechts.

Marketing, Lizenzen.
IT, Technik.
Archiv. Ja, das kennen wir schon.
Zwischenzeit: 15:00
Wie gesagt, für alle Mitarbeiter wird das 'ne ganz einfache Sache sein. Aber ich vermute mal, Controlling. Hmm. Hmm. So komm' ich da also nicht rein. Ah.
Gut.

??? *(war nicht zu verstehen)*

BIRGIT: Ich habe vorhin schon mal gesagt, das ist auch der C F O.
Ja, aber den habe ich noch nicht gefunden.

Da ist er!

So, wie war die Aufgabe noch mal? Wer muss den Urlaubsantrag von Graf Hasimir Fenring unterschreiben? Das ist damit ja dann auch klar, das ist Shaddam IV.

Zwischenzeit: 16:08
Aufgabe 6: Wieviele Mitarbeiter hat die Programmredaktion? Einfach mal zurück.

Pfeift

Blättert in den Aufgabezetteln
Oh, ich bin gleich fertig.

Ah. Ist auch spannend.

Zwischenzeit: 17:00
... Mitarbeiter hat die Programmredaktion.

Programmredaktion.

Eins, zwei, drei.

Da sind keine weiteren. Und da. Büro LA. Oh. Vier!

Nö nö. Ja, dann hat die Programmredaktion wohl drei Mitarbeiter, weil das ist hier ist ja schon wieder was anderes.

Drei!

Endzeit: 18:08

Testteilnehmer 2: Dankward

Startzeit: 0:00
Dann fang' ich an. Dann soll ich jetzt also laut Aufgabe 1, ähm, Informationen über dieses Organigramm suchen, ähm, Moment, ich muss es noch mal lesen. M-mhm.
Zwischenzeit: 0:32
Ein Frau aus der IT-Abteilung.
Dann dreh' ich hier jetzt mal an dem großen Rad und guck mir überhaupt erstmal an ... Das ist offensichtlich unterteilt in drei Hauptstränge. Da will ich ja dann ...
Da geht's ja schon los.
Wo finde ich denn jetzt die IT. Das ist offensichtlich ein Zweig von Finanzen, finde ich. Also, dann gehe ich auf Finanzen, drehe da und sehe da. Ja. Dann im dritten Unterarm die IT, in Klammern EDV. Da klick' ich drauf.
Und da tut sich auf - die EDV. Da seh' ich dann schon ganz viele Bilder auf einmal.
Und dann sollte das, ähm, eigentlich schon getan sein, weil, ja, ich weiß wie diejenige ausgesehen hat, also ich mein' ...
Zwischenzeit: 1:37
Ach so, ich muss ja nun noch den Namen und die Telefonnummer, ähm. Gut, dann klick' ich jetzt. Ich sach jetzt einfach mal. Es gibt hier ja mehrere Frauen. Ich such' mir jetzt hier einfach mal eine aus.
Und da klick ich dann auf das Bild und hab' den Namen und die Telefonnummer. Damit ist das dann gemacht.
Zwischenzeit: 2:06
Dann die nächste Aufgabe. Ich soll herausfinden, wer der direkte Vorgesetzte von Thufir Harwat aus dem Archiv ist.
Dann geh ich erstmal wieder zurück auf's Diagramm.
Archiv. So. Das lese ich dann gleich hier direkt an dem Arm, wo auch die IT dran hängt. Klick ich da drauf.
Zwischenzeit: 2:32
So.
Wobei ... Ich soll ja eigentlich nach einem Namen suchen und nicht nach einem, ich

meine gut, ähm.
Das könnte ich jetzt wahrscheinlich so machen, vielleicht auch irgendwie anders. Das. Tja. Also ähm.
Ich weiß ja jetzt, wer Thufir Harwat ist und seh' jetzt halt schon an der Anordnung der Bilder, dass die direkte Vorgesetzte eben Elke Korol ist, weil sie hier eben, ähm, mittig angeordnet ist. Wobei das dann natürlich auch erstmal dann nachzuprüfen wäre, indem ich halt darauf klick und dann steht da halt Leitung bei ihr.
Zwischenzeit: 3:32
Ja, ich denk' mal, so wird das gehen.
Ja, OK.
Zwischenzeit: 3:47
Ähm, dann ist die dritte Aufgabe: Welches Vorstandsmitglied ist für die Kinoredaktion verantwortlich? Welchen Bereich leitet diese Person?
Hm, Vorstand.
Da guck' ich dann.
Gibt's hier denn nicht irgendwo eine Suchmaske oder so?
Zwischenzeit: 4:32
Vielleicht da? Dann guck' ich einfach mal. Was ist hier.
Ich hab' jetzt einfach mal auf Reality geklickt. Da ist jetzt irgendwie auch das Bild nicht vollständig. *(Siehe Abbildung A.3)*
Wieder zurück.

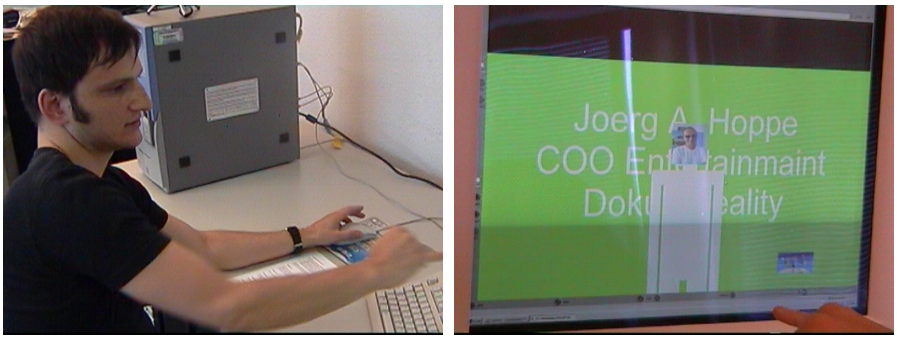


Abbildung A.3: Dankward hat aufgrund des weißen Körpers den Eindruck, dieser sei nicht vollständig.

Dann such' ich jetzt hier einfach mal ein bisschen.
Zwischenzeit: 5:32
Und gucke nach der Kinoredaktion.
Hm.
Reality.

Und wieder zurück.
Zwischenzeit: 6:32
seufzt
Was sind denn das für Buttons, die habe ich mir noch gar nicht angeguckt. Das ist offensichtlich ... Geh ich hier noch mal in die Räumlichkeiten - ach so, das ist hier der Goto-Button, da wird dann der Cursor zu etwas anderem. *(siehe Abbildung A.4)*

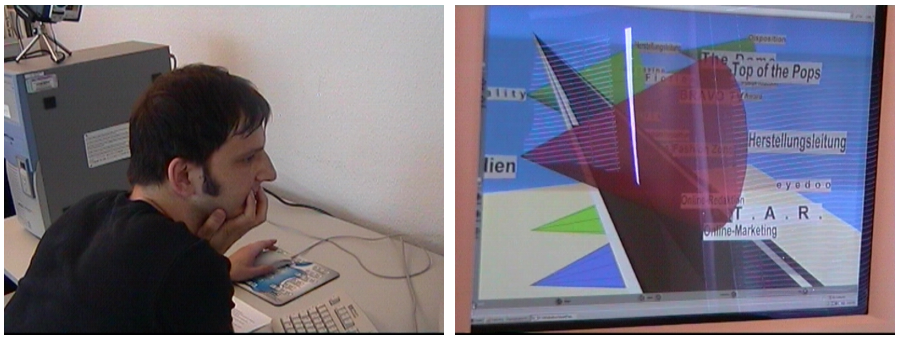


Abbildung A.4: Der schwarze Strich in die Tiefe scheint eine Fehlfunktion des VRML-Viewers zu sein.

Zwischenzeit: 7:32
Wie komme ich denn jetzt wieder zurück? Klick ich Restore ...

Und wieder Restore.
Ich geh mal auf ...
Zwischenzeit: 8:32
Ach so, nee. Wenn ich jetzt auf Goto klick ...

Damit kann man noch ...
Springen. Springen mit den View-Buttons. Aber ...
Da klick' ich jetzt durch offensichtlich die einzelnen Redaktionen, die ich in diesem Rahmen, den ich jetzt gerade angeklickt hab', waren. Das seh' ich in der Übersicht.
Zwischenzeit: 9:32
So, also jetzt hab ich hier natürlich ...
Eine Abteilung, die nennt sich Fiction. Und ich denke mal, dass das die Kinoredaktion sein soll, das geht hier aber irgendwie so richtig nicht hervor.
Ich geh' auf die Übersicht.
Dreh noch mal an dem gelben Rad.
Und sehe dann halt ... Ach so, da sehe ich Kinoredaktion, jetzt. Da klick ich drauf. Das hat aber lange gedauert.

Interessant.
Zwischenzeit: 10:32
Ähm, so, jetzt bin ich in der Kinoredaktion. Jetzt hab ich da eben mittig wahrscheinlich wieder die Leitung angeordnet. Das ist Lady Jessica.
Und.
Jetzt soll ich aber ja herausfinden, welches Vorstandsmitglied ???.
Also seh' ich, das ist ein grüner Arm, das ist Reality. Also ist das Leto Atreides.
Zwischenzeit: 11:23
Und ja, mit Reality ist dann der Bereich ja auch schon genannt.
Zwischenzeit: 11:32
Der gefragt ist.
Das war dann schon wieder richtig gut.
Zwischenzeit: 11:42
Dann kommt die nächste Aufgabe: Welcher Abteilung ist die Disposition untergeordnet. Dann muss ich dann halt hier irgendwie drehen, bis ich die Disposition finde. Das liegt hier auch wieder im grünen Bereich. Das ist untergeordnet, der Herstellungsleitung. Das sehe ich dann auch hier in der Übersicht, aber, na ja.
Zwischenzeit: 12:14
So, dann wär' ... Wer muss den Urlaubsantrag von Graf Hasimir Fenring, sein direkter Vorgesetzter ist also gesucht, unterschreiben.
Zwischenzeit: 12:32

CFO. Ah ja. Ein Graf Hasimir Fenring ist ein CFO. Einen CFO habe ich hier natürlich gefunden. Und der direkte ... Ich klick erst mal bei CFO rein und guck, was ...

Da seh' ich, wofür er zuständig ist, aber ich will ja jetzt den direkten Vorgesetzten wissen. Das wird sicherlich derjenige sein, der im Baum darüber ist, und das geht ab von Finanzen und dann haben wir hier den Vorgesetzten Shaddam IV., Vorsitzender, CEO.
Zwischenzeit: 14:44
So, und dann soll ich herausfinden, wie viele Mitarbeiter die Programmredaktion hat.
Die Programmredaktion...
Zwischenzeit: 15:04
Ende des Bandes.

Testteilnehmer 3: Michael

Startzeit: 0:00
Die IT-Abteilung. Wo haben wir die IT-Abteilung. Dann drehen wir doch einmal. Ah, noch sehe ich sie nicht, aber das kann sich nur um Sekunden handeln. Vielleicht bin ich ja blind? Nein, jetzt habe ich sie gefunden.
Da klicken wir doch mal auf die IT-Abteilung.

Zu wenig geklickt? Ah ja, jetzt kommt was. Jetzt kommen auch Bilder. Ich suche eine Person aus der IT-Abteilung. Wer war denn das? Die Telefonnummer habe ich ja nicht, aber ich kann sie wieder erkennen. Da ist sie, die Birgit. Dann klicken wir doch mal auf die Birgit rauf. Ah, ja. Da ist das, der Steckbrief von der Birgit. Wenn ich jetzt ein Telefon hätte, würde ich sofort die Fragen, die ich habe, loswerden können.
Zwischenzeit: 0:54
Somit habe ich meine Aufgabe 1 hoffentlich erfolgreich beendet.
Zwischenzeit: 1:06
Wie kommen wir wieder zurück? Einmal Kreuz, bekannt. Restore. Und wir haben unser..

Unser Modell, wo bleibt das Modell? Modell? Modell? Da rauf klicken? Jupp. geht auch.
Zwischenzeit: 1:28
Vorgesetzter von Thufir? Archiv. Wo haben wir das Archiv?
Oh, das ist sogar ganz in der nähe von der IT. Dann klicken wir doch mal auf das Archiv.

Das ist aber ein langsamer Rechner.
Da haben wir die drei Mitarbeiter. Wer ist direkter Vorgesetzer von dem Thufir? Wer ist denn jetzt Thufir? Ist das der Linke?
Zwischenzeit: 1:54
Nein, das ist der Fremen. Gehen wir noch mal rechts. Ja das ist der Thufir. Da gibt es keine Informationen, wer der Vorgesetzte ist.
Klicken wir mal auf die nächste Person. Leitung Archiv. Das scheint die Lady Jessica zu sein.
Frage Zwei beendet.
Und zurück zur Ausgangsposition.
Zwischenzeit: 2:34
Welches Vorstandsmitglied ist für die Kinoredaktion verantwortlich? Welche Bereiche leitet diese Person? Kino-Redaktion? *(siehe Abbildung A.5)*
Kino.
Ah, schön wär's, wenn ich jetzt so'n Such-Button hätte.
Zwischenzeit: 2:54
Kino und dann geht's weiter. Aber der ist nicht da. Dann müssen wir so mal gucken.
Postproduction. Redaktion. Hab ich mich verlesen? Kino? Kinorerdaktion.
Tja, vielleicht drehen wir mal, da hinten ist ja noch was.
Kino.
Kino-Redaktion. Da haben wir die Kino-Redaktion.
Zwischenzeit: 3:54
Da haben wir die, aber, da bin ich ja schon zu weit in der Tiefe. Nein, wir wollen nicht Redaktion, wir wollen ja den Chef von dem Ganzen. Dam da dam da da.
CFO? Gucken.

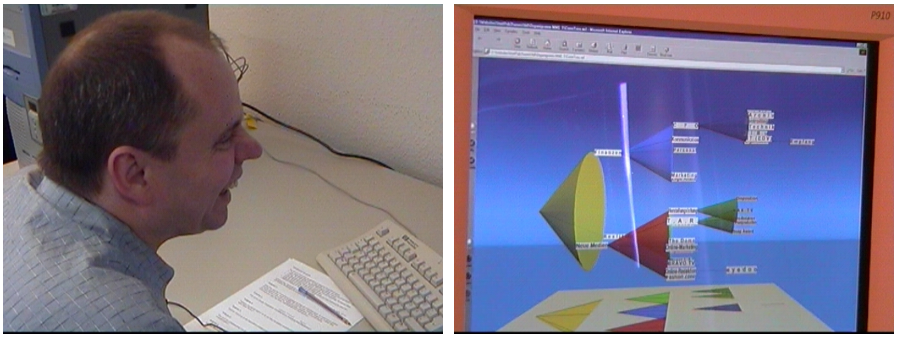


Abbildung A.5: Das vollständige Videointerview mit Michael ist auf der beiligenden CD-ROM zu finden.

Nein. Da würd' ich sagen. Gehen wird doch mal nach ganz oben.

Zwischenzeit: 4:54
Welches Vorstandsmitglied, wo finden wir denn Vorstandsmitglieder?

Ah ja, Leto Atreides.
Ja, hätte ich mir ja fast denken können.

Welchen Bereich leitet diese Position?

Leto Atreides.
Zwischenzeit: 5:54
Welchen Bereich? CEO Entertainment. Docu-Reality. Oh, gut.

Zwischenzeit: 6:05
Frage Nummer 4: Welcher Abteilung ist die Disposition untergeordnet?
Dispo. Hmm, Dispo, wo haben wir die Dispo. Ach, wo ist mein Suchknopf? Nein, zufälligerweise habe ich es gefunden.

Da ist sie.
So lernt man die Leute kennen.

Dispo? Dispo. Herstellungsleitung.

Zwischenzeit: 6:59
Sein direkter Vorgesetzer von... Hasimir. Das ist die Aufgabe 5. Den habe ich doch dann

auch schon irgendwo gesehen. CFO.

Wo haben wir den Vorgesetzten?
Dann müssen wir wieder zurückgehen, das ist der Richtige. Eine Stufe höher. Finanzen. Da da dat. Ein Doppelklick. Der Shaddam. Shaddam muss den Urlaubsantrag von Hasimir unterschreiben.

Zwischenzeit: 7:39
Frage Nummer 6: Wie viele Mitarbeiter hat die Programmredaktion?
Hauptmenü.
Programmredaktion.
Dum dum dum.

Zwischenzeit: 7:54
Hmm-mm.

Dispo.
Was ist denn das da hinten?

Tja, da bin ich jetzt am Suchen. Was ist mit Programmredaktion gemeint? Wo haben wir die? Wo versteckt sich die Programmredaktion?
Hmm-mm-mmm.
Programmredaktion. Da.
Doppelklick.

Zwischenzeit: 8:54
Wer ist denn das? Irulan. Noch nie gesehen.
Die kennen wir auch.
Atreides. Ah ja.
Tja. Wenn das alles ist auf dem Foto, würde ich sagen, drei Mitarbeiter.
Das zur Aufgabe sechs.
Endzeit: 9:24

Testteilnehmer 4: Carsten

Startzeit: 0:00

Und wo soll jetzt die IT sein? Finanzen?

Zwischenzeit: 0:24

BIRGIT: Was siehst du gerade, Carsten?

Seh leider nur die Neuen Medien, aber nur die IT-Welt nicht. Und die such' ich ja.

Da, schau.

IT unter Personal und Finanzen ist ja nicht gerade sinnig.
Zwischenzeit: 1:24

Und gefunden anhand ... Bild.

Zwischenzeit: 1:49
Gut, zweite Aufgabe.

Zurück.

Finanzen.
Zwischenzeit: 2:24

So, einmal Archiv und ... suchen ... Thufir Harwat.

BIRGIT: Welche Aufgabe löst du gerade?
Zwei. *(siehe Abbildung A.6)*

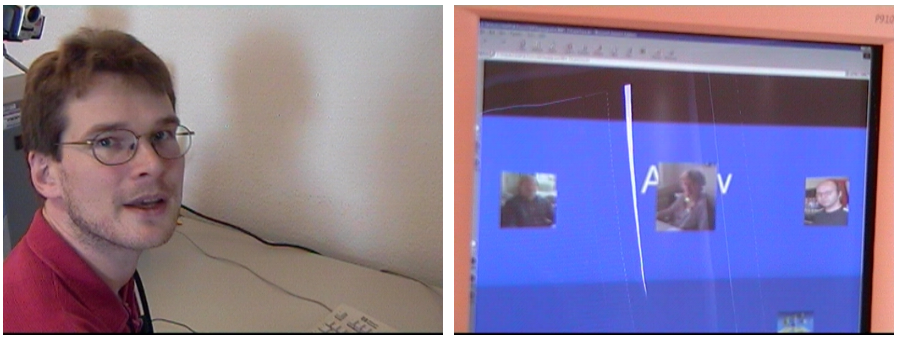


Abbildung A.6: Dieser Teilnehmer ließ sich trotz mehrmaliger Aufmunterung nicht zum Sprechen ermutigen.

So, Lady Jessica ist die Leiterin Archiv. Mal seh'n ob auch beim Thufir so ...

Na gut.
Zwischenzeit: 3:09
Jetzt zurück und Aufgabe 3.

Zwischenzeit: 3:24
Kino-Redaktion.

Ist das normal, dass sich das weiterdreht?
BIRGIT: Kommt durch die Verzögerung.
Ach so.

Zwischenzeit: 4:24

BIRGIT: Und was machst du jetzt?
Das ist ganz schön (*ist ironisch gemeint*). Wär' schön, wenn man irgendwie zurückspringen könnte, wie beim Internet Explorer. Jetzt bin ich zwar einmal zurückgegangen, nur, jetzt muss ich ganz von vorne anfangen zu suchen.

So. Vorstandsmitglied Kinoredaktion. Welcher Bereich.

Dann gucken wir unter Reality.
Chef ist Leto Atreides.
Zwischenzeit: 5:24

Das ist nicht schön.

Wäre schön, wenn er auf dem letzten Bild zurückgeblieben wäre. Na gut.
Zwischenzeit: 5:56
Dann Aufgabe 4.

Welche Abteilung. Disposition? Höchstwahrscheinlich Finanzen.
Nein.

Zwischenzeit: 6:24

Da schau'n unter neue Medien. Herstellungsleitung, Disposition.

Untergeordnet der Herstellungsleitung.
Zwischenzeit: 7:04
Gut, Aufgabe 5.

Zwischenzeit: 7:24
Fenring. CFO. Finanzen.

Sein direkter Vorgesetzter sollte sein ...
Shaddam IV. Und der ist es.

Zwischenzeit: 8:01
Gut, Aufgabe 6.
Wie viele Mitarbeiter hat die Programmredaktion?

Zwischenzeit: 8:24

Da, unter Reality.

Gut, wenn man das so richtig sieht, drei Mitarbeiter.
Vier.
Fünf.
Sechs, sieben acht. *(siehe Abbildung A.7)*

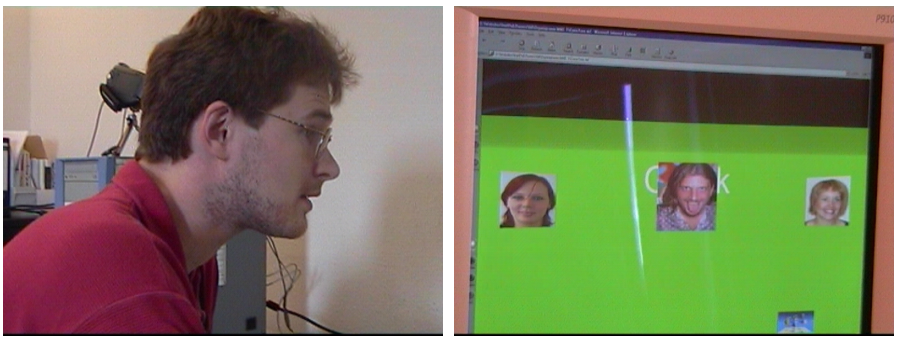


Abbildung A.7: Immer mehr Mitarbeiter in der Abteilung Programmredaktion? Der Teilnehmer zieht erstaunt die Augenbrauen hoch.

Nein, das ist wieder die Kino-Redaktion. Das kann nicht sein.

So nicht.
Zwischenzeit: 9:24

BIRGIT: ??? (war nicht zu verstehen)
Einfach die Pfeiltaste. Die Überlegung war einfach, da das eine Bild nur halb drauf ist auf der Seite einen Tick nach rechts zu schwenken.
Nur darunter landet man dann wieder in anderen Redaktionen.
Also hier sind, seh ich, das jetzt drei ...
Irgendwie die Pfeiltaste, um nur einfach zu drehen.

Noch mal nach rechts.
Doch, jetzt funktioniert's.
Zwischenzeit: 10:24
Tatsächlich nur drei.
Zwischenzeit: 10:30

Testteilnehmer 5: Franz

Startzeit: 0:00
Also, eine Frau aus der IT-Abteilung realisiert für dich ein Intranet- Projekt, doch nun fallen dir noch ein paar Fragen ein. Leider hast du ihren Namen und ihre Telefonnummer vergessen. Du suchst deine Informationen über das neue, dreidimensionale Organigramm. Gut, dann will ich das doch mal tun. Dann drehe ich erst mal ganz oben, natürlich.
Zwischenzeit: 0:28
Äh, düdellütelütelüt. Das wär' vermutlich, jetzt nehm ich mal an, Finanzen könnte das sein. Geh ich mal einfach von aus. Gut.
Da gibt's Personal, das ist ja nicht EDV, Legal Affairs ist auch nicht EDV, Marketing auch nicht, Kommunikation auch nicht, CFO könnte es sein, guck ich mal. Yeah, Mann, there is it. Dann geh ich mal auf die EDV. Dat düert erst mal.
Oh, Mensch, da seh' ich ganz viele Bilder.
Hmm-mh. Entscheide mich mal, das ist ein Mann, das ist auch ein Mann, das ist auch ein Mann, das ist eine Frau, die könnte das sein. Nee, die hatte blonde Haare, das ist die da.
Zwischenzeit: 1:28
Gut, die sitzt in Raum 1-0-8 und hat die Telefonummer 7-4-5. Und heißt Madame Labouche.
So.
Das schreib ich mir noch mal auf, ne, 7-4-5. Und Raum 1-0-8, nee. Gut. Und dann stand da oben ja irgend was mit Fenster zumachen. Try to close the window? Yeah, I want to close the window *(siehe Abbildung A.8)*. Komme ich da vermutlich zurück. Yeah. Gut.
Zwischenzeit: 2:06
Wer ist der, also zweite Aufgabe, fand ich schon gut jetzt, ne. Ging ziemlich fix. Ich bin halt genial. Wer ist direkter Vorgesetzter von Thufir Harwat, Abteilung Archiv. Gut, dann gucke doch auch noch mal.
Zwischenzeit: 2:28
Öh, Archiv gehört nicht zu Neue Medien, nicht zu Reality, vermutlich auch CFO, ja, das ist es ja schon. Mal sehen. Yeah. There is Archiv.
Pfeift
Da ist Thufir, nehme ich an, Leitung Archiv. Wunderbarst.

Schwupp.

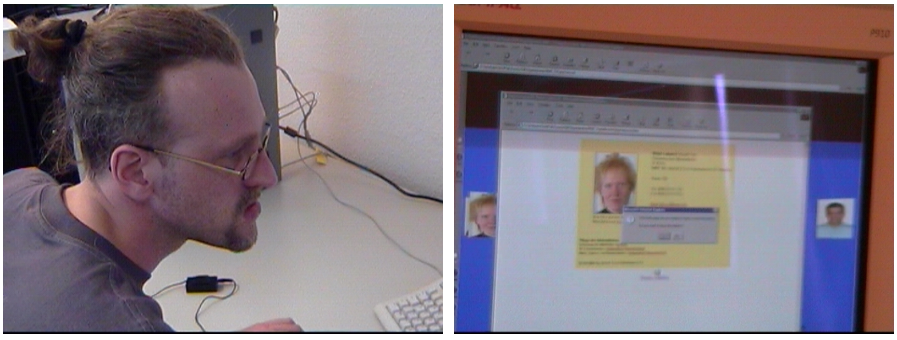


Abbildung A.8: Franz schließt ein Mitarbeiter-Profil.

Zwischenzeit: 4:14
Welches Vorstandsmitglied ist für die Kinoredaktion verantwortlich, welchen Bereich leitet diese? Düddelüddellüt.
Da muss ich dann ja mal nach Kino suchen.
Zwischenzeit: 4:28
Postproduction, Disposition ist es nicht, ne. Ist es der? Nee.
Da.

Dat war Sample Tracks, Soap. Das ist die Kino-Redaktion. Warte mal.
Vorstandsmitglied. Gut. Dann muss man ja, also der Vorstand von Reality. Das ist gerade ein bisschen doof zu sehen, find' ich hier.
So, Kino gehört zu Magazine. Und Magiazine gehört zu Reality. Und Reality ist das da.
Mal gucken, was passiert.
Mensch, Du.
Zwischenzeit: 5:28
Meister Atreides.

Das fand ich jetzt ein bisschen assig zu sehen.
Kann man das auch eigentlich anders rum machen, das wär' ja ganz schlau?
Wart' mal. Reality.

Müssen wir mal auf Magazin gucken.

Was hat die da zu suchen.
Da komm' ich nicht weiter.
Klick ich mal hier hin.
Zwischenzeit: 6:28
Pfeift

Das war Tracks, Soap, Kino. Ja, wunderbar, das ist Atreides.

Zwischenzeit: 6:50
Welcher Abteilung ist die Disposition untergeordnet?
Disposition ist da.
Nee, Reality ist das auch.

Herstellungsleitung würde ich mal sagen.
Hmm-mm. Oh Mensch, jo.

Zwischenzeit: 7:33
Wer muss der Urlaubantrag von Hasimir Fenring, unterschreiben?

Da-ha.
Ist Chief Financial, ist da.
Hm, mmh, mmm.
Ist das schlau, das Bild.
Captain Shaddam.

Zwischenzeit: 8:22
Und wie viele Mitarbeiter hat die Programmredaktion.
Zwischenzeit: 8:28

??? *(war nicht zu verstehen)*

??? *(war nicht zu verstehen)*

Aha.

Da ist sie doch.

Huch, was ist das? Align. *(siehe Abbildung A.9)*

Das ist ne gute Frage. Warte mal, da krieg ich ja wieder nur das Bildchen.
Zwischenzeit: 9:28
Eins, zwei. Das ist...
Das da, denn nehm ich. Die guckt ja auch schlau drein.
Zwischenzeit: 9:36
Ende des Bandes.

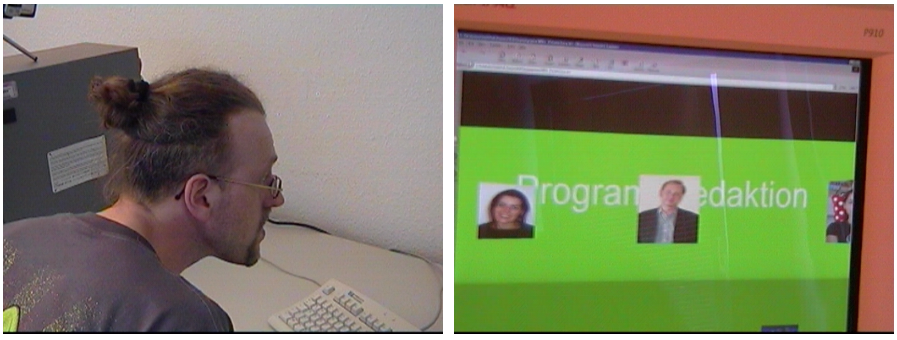


Abbildung A.9: Franz untersucht die Buttons des VRML-Viewers.

Testteilnehmer 6: Christian

Startzeit: 0:00
Mensch, wie hieß denn noch die Frau, mit der ich dieses Gespräch hatte über das tolle neue Intranet-Projekt? Hm, Mensch, guck ich doch mal im Organigramm nach.
IT, IT, da war's doch. Klick ich mal darauf. Mal gucken, wer da alles arbeitet.
Oj, das sind aber viele.
Christian fehlt ja.
Oh nein, hier ist er ja.
Zwischenzeit: 0:47
Ah, Birgit Labusch die war es, glaub ich.
Ja.
Genau, jetzt habe ich sie.
Gut.

Zwischenzeit: 1:00
Wer ist der direkte Vorgesetzte von Thufir Harwat, oder von, oh Gott, Thufir Harwat. Wieder zurück. Thufir Harwat, Archiv, Archiv, Archiv. Das ist doch auch irgendwo hierbei. Da ist Archiv, im selben Baum.
Archiv.

Nee, der ist nur Archivar. Das ist Thufir. Und wer ist sie? Leitung Archiv!
Lady Jessica.Also Lady Jessica, würde ich mal so sagen.
Zwischenzeit: 1:50
Welches Vorstandsmitglied ist für die Kinoredaktion verantwortlich, welchen Bereich leitet diese Person?
Kino-Redaktion? Wo ist denn die?

Ah, da unten.
Nach vorne holen.

Nee, da ist es. Ah si.
Kino-Redaktion. Welches Vorstandsmitglied ist verantwortlich?. D.h. ich geh den Baum hoch und irgendwo hier in Reality würde ich mal sagen ist der Bereich.
Und das ist Leto Atreides. COO Entertainment, Docu und Reality.
Documentation and Reality. Damit ist die Frage wohl beantwortet.
Zwischenzeit: 3:00
What else. Ähm, welcher Abteilung ist die Disposition untergeordnet?
Disposition. Disposition.
Hee? *(siehe Abbildung A.10)*

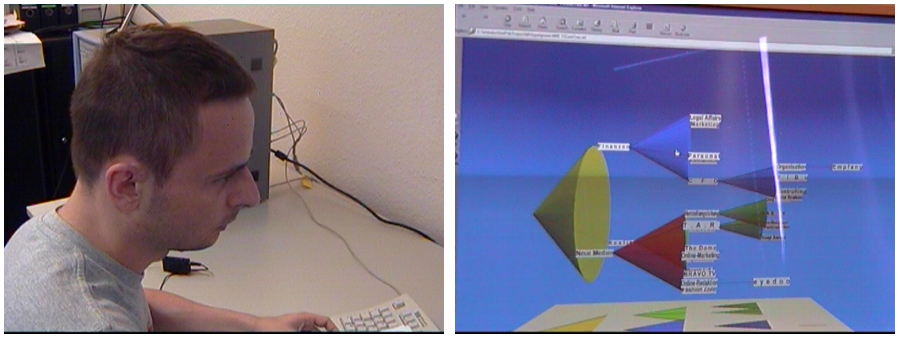


Abbildung A.10: Christian stehen „Fragezeichen“ in's Gesicht geschrieben.

Ach, hier ist sie.
Im grünen Bereich.
Da ist die Disposition.
Welcher Abteilung ist die Disposition untergeordnet?
So wie es hier aussieht der Herstellungsleitung. Mal gucken.
Ja, Herstellungsleitung.
Zwischenzeit: 3:45
Aufgabe 5: Wer muss den Urlaubsantrag von Graf Hasimir Fenring, CFO, unterschreiben?
Hier ist CFO. Blau.
Ja, OK, das ist er. Geh'n wir noch mal zurück.
Sein direkter Vorgesetzter ist dann. Ach so, das ist jetzt hier oben.

Finanzen? Shaddam IV! Vorsitzender, COO. Würde ich mal vermuten.
Zwischenzeit: 4:25
Aufagbe 6: Wie viele Mitarbeiter hat die Programmredaktion? Wieder in den Baum zurück.
Hmm, Programmredaktion.
Wo ist denn die?

Online-Redaktion, Online-Marketing.
Zwischenzeit: 4:47
Nee, hier ist sie nicht. Hm.
Da ist Programmredaktion.
Ganz nach vorne.
Wie viele Mitarbeiter? Gehen wir doch mal da rein.
Programmredaktion.

Rechts? Hm, sind das schon alle hier?
Kann man sich einmal im Kreis drehen? Oh ja. Oi. *(siehe Abbildung A.11)*
Also, da sind dann insgesamt nur drei.

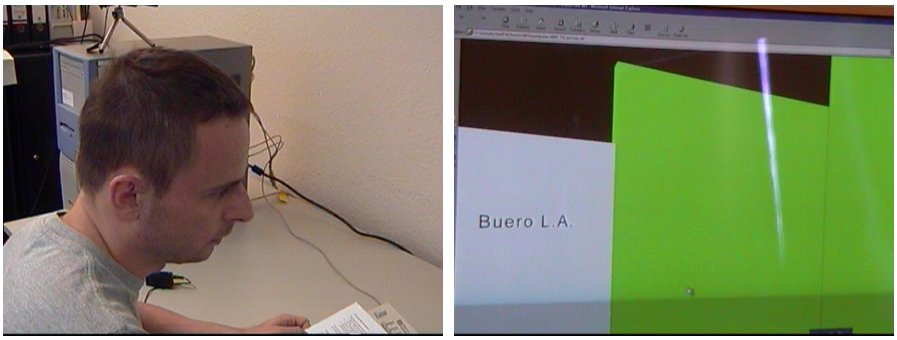


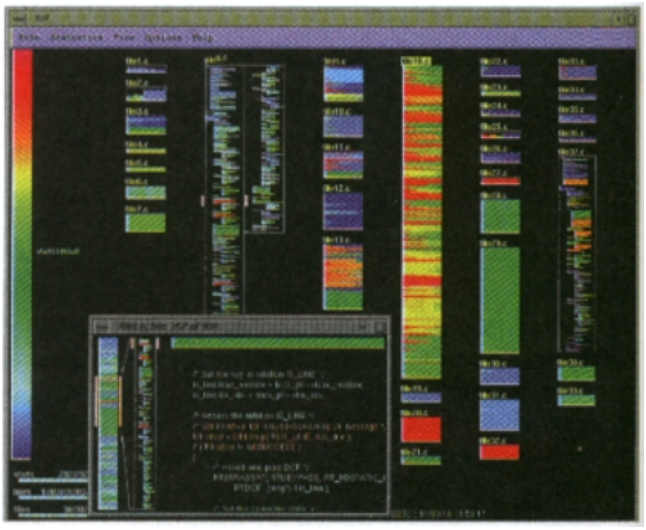
Abbildung A.11: Christian navigiert im Gebäude.

Drei Mitarbeiter.
Wer ist er hier eigentlich? Dr. Keynes. Hab' ich ja noch nie gesehen.
Drei Mitarbeiter hat sie.
Das war's.
Endzeit: 6:16

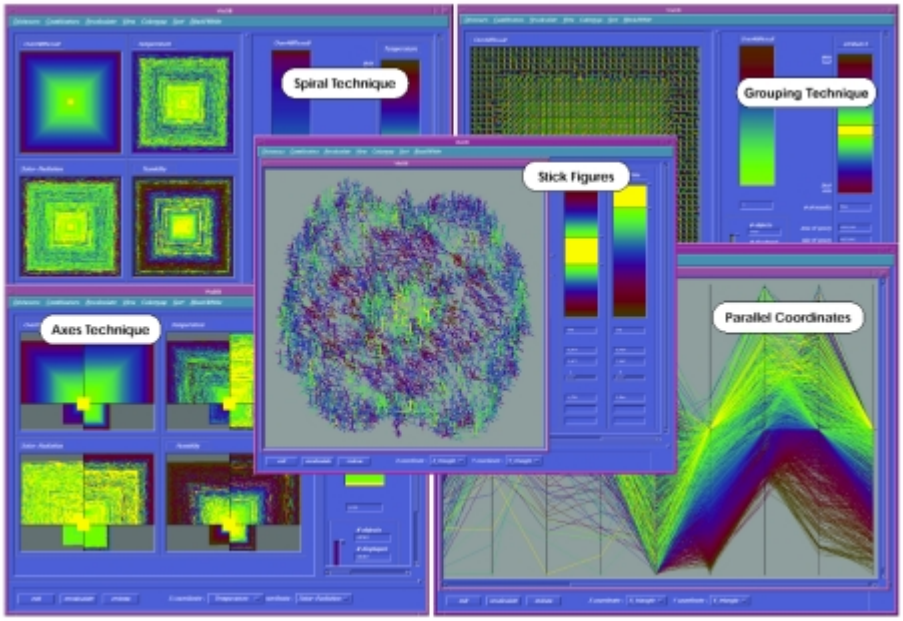
Anhang B

Farbtafeln

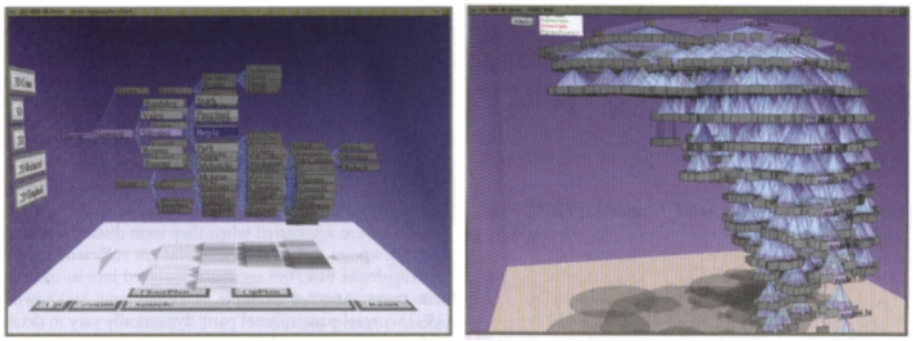
Farbtafel 1: Auswahl einiger Symbole des Star Tree Designers



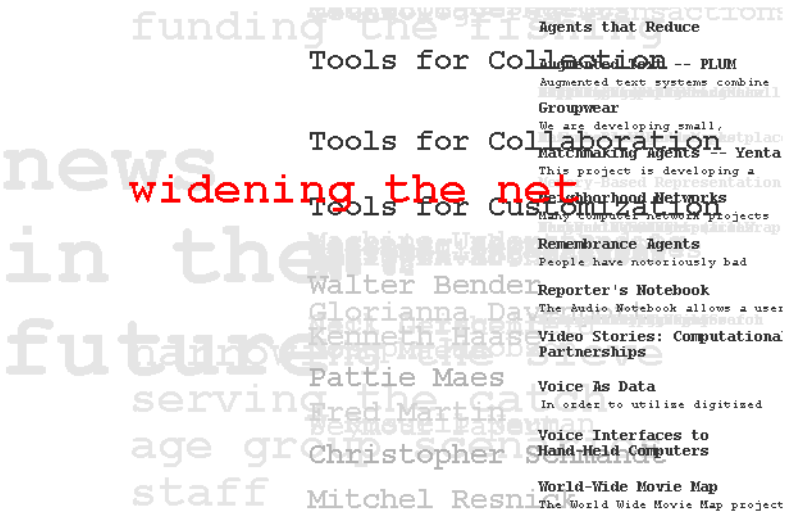
Farbtafel 2: SeeSoft



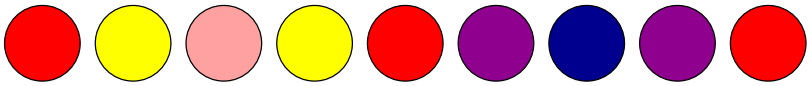
Farbtafel 3: Verschiedene Bildschirmansichten von VisDB



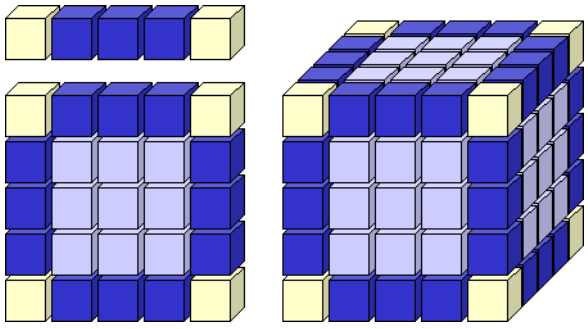
Farbtafel 4: Cone-Tree



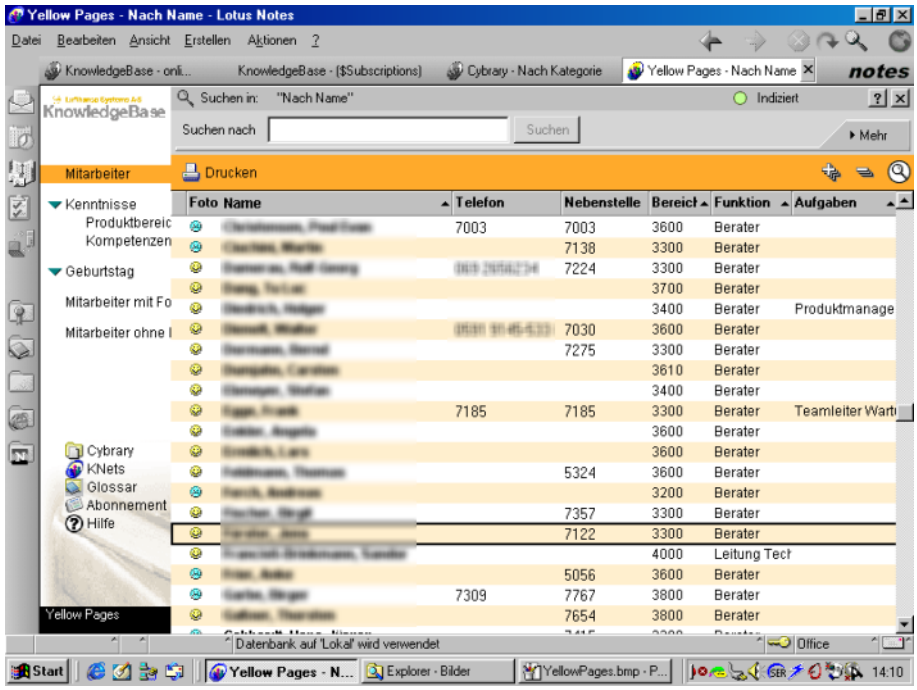
Farbtafel 5: Elastic Catalog



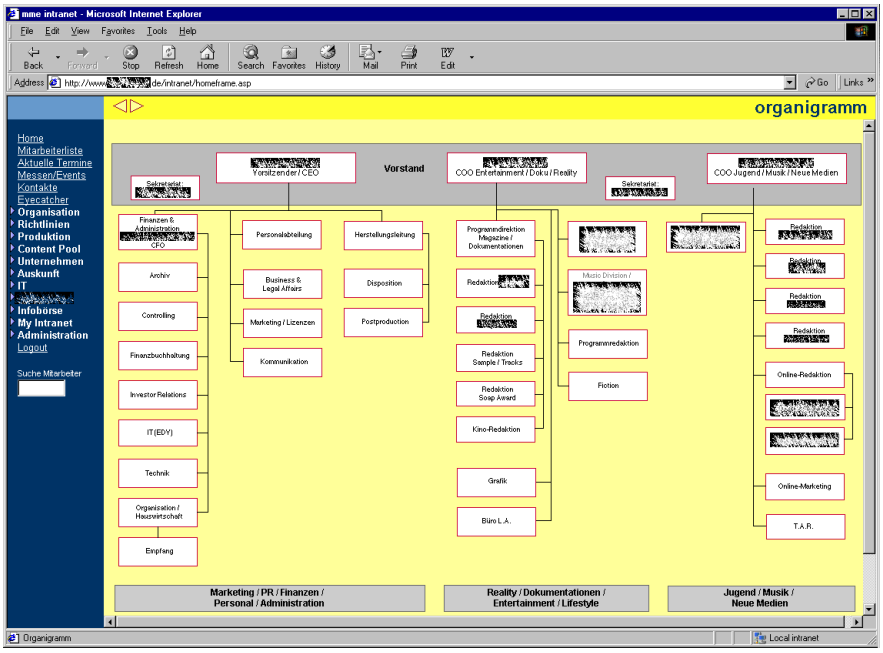
Farbtafel 6: Die Werte einer Sinuskurve



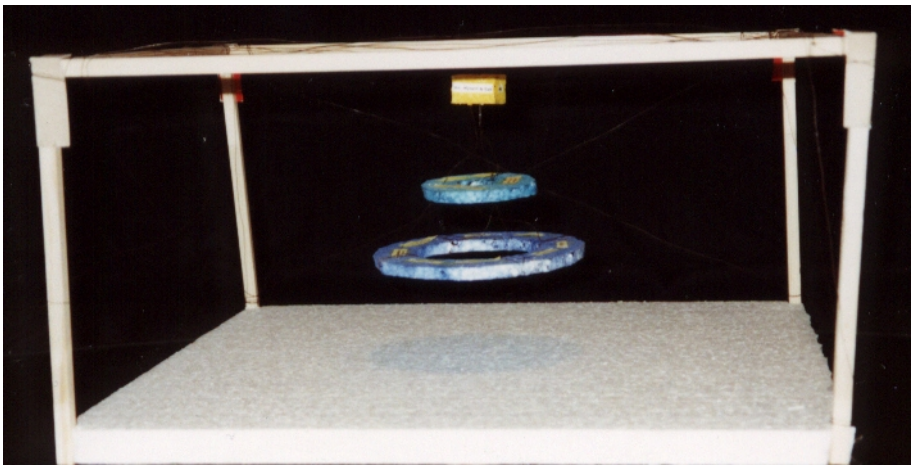
Farbtafel 7: Besondere Orte in der Datenstruktur Feld



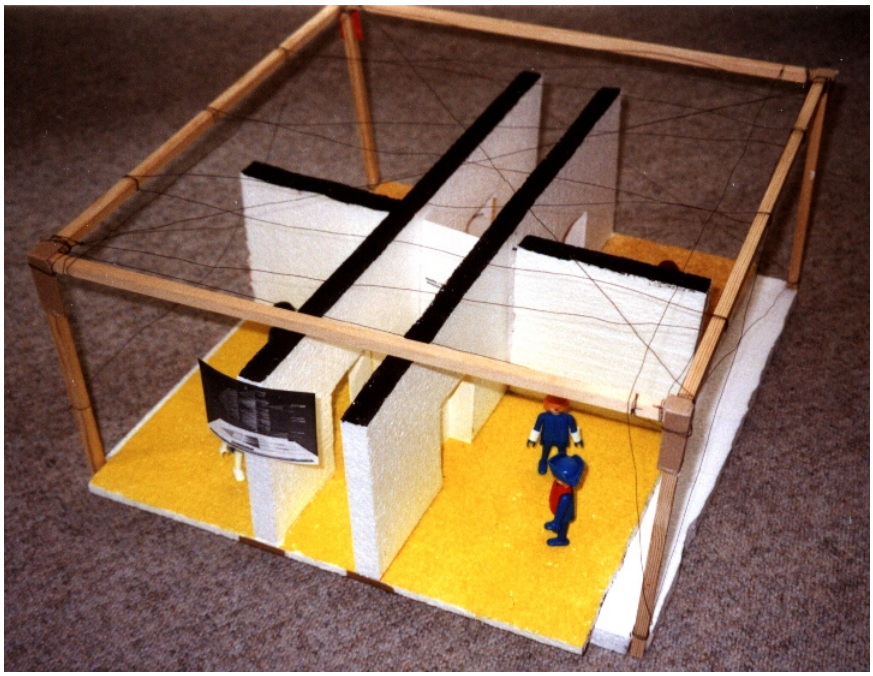
Farbtafel 8: Die *Yellow Pages*



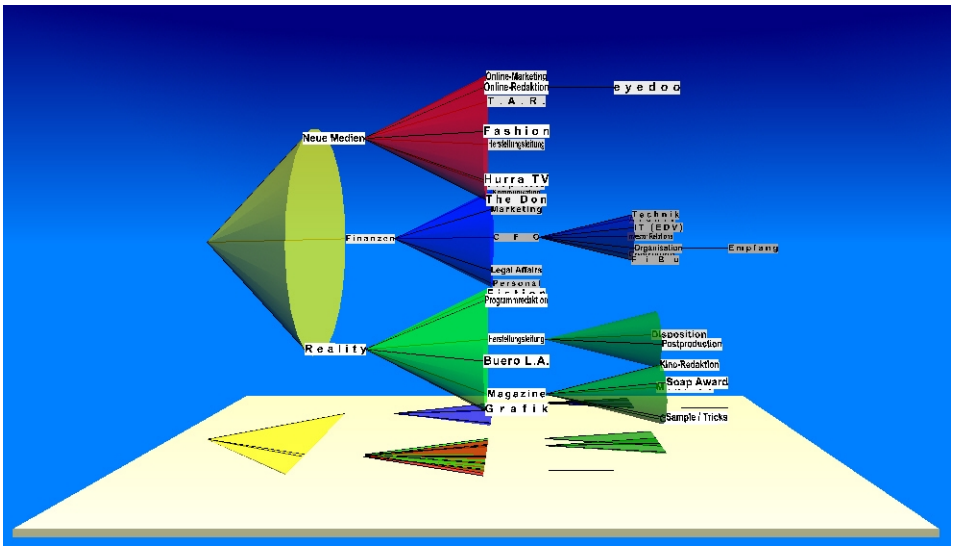
Farbtafel 9: Das Organigramm der `Fernseh AG`.



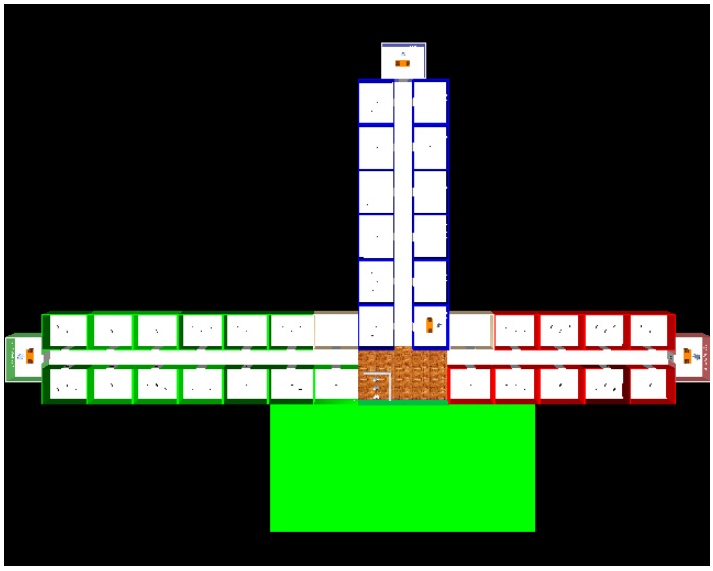
Farbtafel 10: Der Cone-Tree



Farbtafel 11: Ansicht der Räume aus der Vogelperspektive

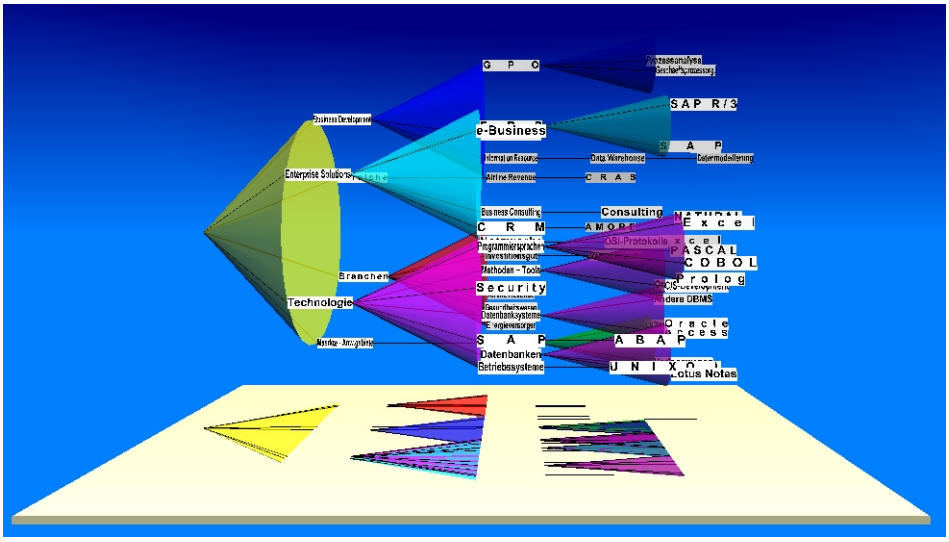


Farbtafel 12: VRML-Cone-Tree mit den Abteilungen der `Fernseh AG`



Farbtafel 13: Grundriss des Gebäudes.

Farbtafel 14: Die IT-Abteilung der `Fernseh AG`.



Farbtafel 15:
VRML-Cone-Tree mit den Kenntnisbereichen der `Lufthansa Systems AS`.

Hamburg, den 29. August 2001

Ich versichere hiermit, die vorstehende Arbeit selbständig und unter ausschließlicher Zuhilfenahme der angegebenen Hilfsmittel erstellt zu haben.

Birgit Labusch