

Entwicklung einer Querycollection für das Hyper-G Authoring Tool Amadeus

Diplomarbeit in Telematik
1997

Fessler Christian Erich

Betreuer: Univ.Ass. Dipl.Ing. Thomas Dietinger

Begutachter: o.Univ.-Prof. Dr. Phil. Dr. h.c. Hermann Maurer

Institut für Informationsverarbeitung und Computergestützte Neue Medien (IICM)
Technische Universität Graz

Zusammenfassung

Diese Diplomarbeit beschäftigt sich mit der Entwicklung einer leicht verständlichen und effizienten Methode zur Speicherung und Abfrage großer Datenmengen, die in einer strukturierten Form vorliegen.

Es wird auf das Internet, dessen Organisation und Mechanismen eingegangen. Es folgt eine Beschreibung des Hyperwave-Information-Management-Systems. Danach wird die Theorie der Mensch-Maschine-Kommunikation behandelt und wie eine praktische Implementation eines User Interfaces zu erfolgen hat. Der Suche nach Information (information retrieval) ist ein weiteres Kapitel gewidmet. Schließlich erfolgt eine genaue Erläuterung der „Querycollection“, deren Anwendung, Vorteile und Implementation.

Abstract

This thesis is about the development of an understandable and efficient method for storing and retrieving huge amounts of structured data.

It deals with the Internet, its organisation and mechanisms. Then there is a description of the Hyperwave-information-management-system. The theory of human computer interaction and the possibilities of information retrieval systems are discussed. At last there is an exact explanation of the so called „Querycollection“ and their usage and implementation.

Erklärung der Selbständigkeit

Ich versichere hiermit, diese Arbeit selbstständig verfaßt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient zu haben.

INHALTSVERZEICHNIS

<i>1</i>	<i>Einführung</i>	<i>9</i>
<i>2</i>	<i>Internet</i>	<i>10</i>
2.1	Entwicklung des Internet	10
2.2	Organisationsstrukturen im Internet	13
2.2.1	Internet Society (ISOC)	13
2.2.2	IAB, IETF und IRTF	13
2.2.3	RARE und RIPE	14
2.2.4	Schaffung von Internet-Standards	14
2.2.5	Internet Assigned Numbers Authority (IANA)	15
2.2.6	Network Information Center (NIC)	15
2.3	Übertragungsmechanismen	15
2.3.1	Kommunikationsprotokolle im Internet	15
2.3.2	TCP/IP	17
2.3.3	Internet-Adressen	19
2.3.4	Domain Name System (DNS)	21
2.4	Dienste	22
2.4.1	Email	22
2.4.2	ftp (file transfer protocol)	23
2.4.3	telnet (remote login)	23
2.4.4	Usenet News	23
2.4.5	World Wide Web (WWW)	23
<i>3</i>	<i>Hyper-G (Hyperwave)</i>	<i>27</i>
3.1	Die Entstehung von Hyperwave	27
3.2	Konzept von Hyperwave	28
3.2.1	Client-Server Architektur	28
3.2.2	Kooperierende Prozesse	28
3.2.3	Datenmodell	29
3.2.4	Links in Hyperwave	31
3.2.5	Suchmöglichkeiten in Hyperwave	31
3.2.6	Benutzerverwaltung und Zugriffsrechte	32
3.2.7	Meta-Information (Attribute)	33
3.2.8	Mehrsprachigkeit	33
3.3	Wavemaster	34
3.4	Hyperwave Publishing Wizard	34

3.5	Hyper-G Client Amadeus	35
3.5.1	Collection Browser	35
3.5.2	Viewer	35
4	<i>Mensch-Maschine-Kommunikation</i>	36
4.1	Einleitung	36
4.1.1	Was ist Mensch-Maschine-Kommunikation (Human Computer Interaction)?	36
4.1.2	Warum Mensch-Maschine-Kommunikation?	36
4.1.3	Mensch-Maschine-Kommunikation als fächerübergreifendes Forschungsgebiet	37
4.1.4	Was ist ein Benutzerinterface?	38
4.1.5	Ziele und Rollenverteilungen bei der Entwicklung von Benutzerinterfaces	39
4.1.6	Phasen der Softwareentwicklung	40
4.2	Meilensteine in der Mensch-Maschine-Kommunikation	41
4.3	Menschliche Informationsverarbeitung	41
4.3.1	Informationsspeicherung beim Menschen	41
4.3.2	Kognitive Modelle in der Mensch-Maschine-Kommunikation	42
4.3.3	Wahrnehmung und Repräsentation von Objekten	42
4.3.4	Aufmerksamkeit und Beschränkungen bei der Informationsverarbeitung	43
4.3.5	Wissen und mentale Modelle	44
4.3.6	Interfacemetapher und Begriffsmodelle	47
4.3.7	Lernen in Zusammenhängen	49
4.4	Gestaltung von Benutzerinterfaces	51
4.4.1	Allgemeines	51
4.4.2	Unterschiedliches Design für unterschiedliche Anwendungen	54
4.4.3	Beachtung der Verschiedenheit der Benutzer	55
4.4.4	Hauptinteraktionsstile beim Umgang mit Computern	56
4.4.5	Regeln für das Design von Benutzerinterfaces	60
4.5	Designmethodik	66
4.5.1	Ein effektives benutzerbezogenes Design beinhaltet mehrere einzelne Phasen, den sogenannten Designzyklus:	66
4.5.2	Möglichkeiten zum Testen der Verwendbarkeit eines Produktes (usability testing)	67
5	<i>Suchen nach Information (Information Retrieval)</i>	69
5.1	Was ist Information Retrieval (IR)?	69
5.2	Grundidee	69
5.3	Begriffsdefinitionen	70
5.4	Klassifizierung von Information-Retrieval-Systemen	72

5.5	Beurteilung der Güte eines IRS	74
5.5.1	Maßzahlen	74
5.5.2	Berechnung der Maßzahlen	75
5.6	Deskriptorenauswahl	77
5.6.1	Manuelle Auswahl	77
5.6.2	Automatische Auswahl	77
5.6.3	Probleme bei der Deskriptorenauswahl	78
5.6.4	Speicherung der Deskriptoren (Methode des inverted index)	78
5.6.5	Maßzahlen zur Gewichtung von Deskriptoren	79
5.6.6	Zusammenfassung	81
5.7	Auswahl mit Rückkopplung (relevance feedback)	82
5.7.1	Gewichtung der selektierten Dokumente	82
5.8	Verwendungsarten von Thesauri	83
5.8.1	vordefinierte Listen	83
5.8.2	Synonymwörterbuch	83
5.9	Erkennung physisch ähnlicher Zeichenketten	84
5.9.1	Unterschiedliche Eingaben aufgrund sprachlicher Gegebenheiten	85
5.10	Vergleich von Zeichenketten	87
5.10.1	Straight Forward (SF) - Algorithmus	87
5.10.2	KMP-Algorithmus	87
5.10.3	BM-Algorithmus oder Algorithmus von Boyer und Moore	88
5.10.4	AC-Algorithmus	88
5.11	abschließende Bemerkungen	88
6	<i>Querycollection für Hyperwave</i>	89
6.1	Motivation	89
6.2	Vergleich Querycollection mit einer normalen Collection in Hyperwave	89
6.3	Typischer Verlauf der Erstellung und Benutzung einer Querycollection	90
6.3.1	Erstellung	90
6.3.2	Benutzung	90
6.4	Anwendungsbeispiele	91
6.4.1	Bibliotheksdatenbank	91
6.4.2	Personendatenbank	91
6.4.3	Audio/Videodatenbank	91
6.5	Queryform-Wizard	91
6.5.1	Was ist ein Wizard?	91

1	EINFÜHRUNG	8
6.5.2	Ziel des Queryform-Wizards	92
6.5.3	Funktionsweise	92
6.6	Eingabe der Metainformation in Amadeus	101
6.7	Suche und Suchergebnisse	102
6.8	Programmierung	103
6.8.1	Queryform-Wizard	103
6.8.2	Enter/Modify Keywords in Amadeus	103
6.8.3	CGI-Script für die Suche	103
7	<i>Referenzen</i>	<i>104</i>

1 EINFÜHRUNG

Diese Diplomarbeit beschäftigt sich mit der Entwicklung einer leicht verständlichen und effizienten Methode zur Speicherung und Abfrage großer Datenmengen, die in einer strukturierten Form vorliegen.

Das Kapitel 2 beschreibt die geschichtliche Entwicklung des Internets, dessen Organisationsstrukturen und Übertragungsmechanismen. Weiters wird auch auf die verschiedenen, existierenden Informations- und Kommunikationsdienste eingegangen. Dabei wird das WWW (World Wide Web) einer genaueren Betrachtung unterzogen.

Im Kapitel 3 wird auf das zur Zeit mächtigste Informationssystem im Internet eingegangen: Hyperwave. Es werden das zugrundeliegende Konzept und die wichtigsten Möglichkeiten und Vorteile vorgestellt. Es wird auch auf die verschiedenen Schnittstellen zum Hyperwave-Server eingegangen.

Das Kapitel 4 befaßt sich mit der Theorie der Mensch-Maschine-Kommunikation, wie Menschen Informationen verarbeiten und wie eine gute Gestaltung von Benutzerinterfaces zu erfolgen hat.

Der Suche nach Information ist das 5. Kapitel gewidmet: Es wird die Grundidee des Information Retrieval vorgestellt, die möglichen Klassifizierungen von Information-Retrieval-Systemen und die Beurteilung deren Qualität. Weiters wird die Problematik der Deskriptorenauswahl, die Auswahl durch Rückkopplung (relevance feedback), die Verwendung von Thesauri und der effiziente Vergleich von Zeichenketten diskutiert.

Im 6. Kapitel wird der Terminus der „Querycollection“ genau erläutert, dessen Unterschied zu einer normalen Collection in Hyperwave und der typische Verlauf einer Erstellung und Benutzung dieser Querycollection. Zusätzlich werden noch einige Anwendungsbeispiele vorgestellt und auf die genaue Funktionalität des „Queryform-Wizards“ zur Erstellung einer Querycollection eingegangen.

2 INTERNET

2.1 Entwicklung des Internet

Ende der Fünfziger Jahre erhielt **RAND**, eine Institution in Santa Monica, Kalifornien den Auftrag ein militärisches Kommando- und Überwachungsnetzwerk zu entwerfen, welches auch noch nach einem feindlichen Atomschlag funktionieren sollte, d.h. es sollte ein dezentrales Kommunikationssystem entwickelt werden, daß auch unter schlechten Bedingungen arbeiten kann [Kyas 94].

Als Reaktion auf den Start des Sputnik-Satelliten der Sowjetunion **1957** wurden von der US-Regierung verschiedenste Forschungsprogramme gestartet mit dem Ziel, die militärische Technologie der USA in eine weltweit führende Position zu bringen.

Unter anderem wurde die **Advanced Reseach Projects Agency (ARPA)** gegründet. Eine der wesentlichen Ziele der ARPA war es, Methoden zur zuverlässigen Datenübertragung über weite Strecken zu entwickeln. Die bisher angewandten Techniken beruhten auf der leitungsorientierten Datenübertragung (Circuit Switch) und waren für militärische Anwendungen nicht zuverlässig genug.

Bei der **leitungsorientierten Übertragung** ist der Aufbau einer physikalischen Verbindung zwischen Sender und Empfänger notwendig, bevor die Daten überhaupt übertragen werden können. Wird diese Leitung aufgrund äußerer Einflüsse unterbrochen, wird damit auch die Datenübertragung gestört. Ein Beispiel für leitungsorientierte Übertragung ist das analoge Telefonnetz. Zwischen den beiden Teilnehmern wird immer eine physikalische Verbindung aufgebaut. Aufgrund dieser Nachteile wurde von der ARPA auf die von Paul Barans bei RAND entwickelte **paketorientierte Datenübertragung** zurückgegriffen. Diese Übertragungsmethode ist wesentlich weniger störanfällig in der Übertragungsinfrastruktur und sollte in den kommenden Jahren die Datenkommunikation revolutionieren.

Bei der paketorientierten Datenübertragung werden die zu versendenden Daten in einzelne Datenpakete aufgeteilt. Jedes Paket wird für sich und unabhängig von allen anderen über das Netzwerk verschickt. Bei jedem Paket wird zusätzlich zu den Daten die Adresse des Senders und des Empfängers und auch die Sequenznummer übertragen. Die Sequenznummer dient dazu, daß die empfangenen Datenpakete wieder in der richtigen Reihenfolge zur ursprünglichen Information zusammengesetzt werden können. Der Transport der Pakete über das Netzwerk wird von spezialisierten Computern (Routern) gesteuert, welche entscheiden, wie das Paket optimal zum Empfänger gelangen kann. Es ist durchaus möglich, daß die einzelnen Pakete ein und derselben

Übertragung sehr unterschiedliche Wege zurücklegen können. In weiterer Folge bedeutet das aber auch, wenn eine Übertragungstrecke unterbrochen wird, werden die Pakete über eine andere Strecke transportiert. Fehlerhafte Pakete, oder Pakete die nicht am Empfänger eintreffen, werden automatisch erneut angefordert und übertragen. Dieses Prinzip der paketorientierten Datenübertragung ist wesentlich weniger fehleranfällig.

Im September **1969** wurde von ARPA das erste experimentelle, auf paketorientierter Datenübertragung basierende Netzwerk (**ARPANET**) in Betrieb genommen. Dieses erste Netzwerk bestand aus 4 Knoten (University of California at Los Angeles, SRI-International Menlo-Park Kalifornien, University of Utah und University of California at Santa Barbara) die untereinander über Telefonleitungen verbunden waren.

Ein Jahr später (**1970**) schlossen sich die Universitäten Harvard und MIT an das Netzwerk an.

1971 befanden sich schon über 30 Knoten im ARPANET. Das Kommunikationsprotokoll des ARPANET war zwar paketorientiert, hatte aber mit dem heute verwendeten Übertragungsprotokoll (TCP/IP) noch nichts gemeinsam.

1973 begann die inzwischen in **DARPA** umbenannte Defense Advanced Research Projects Agency ein Projekt, um die unterschiedlichen Implementierungen von paketorientierten Übertragungsmechanismen, die entstanden sind, miteinander zu verbinden: das **Internet-Projekt**.

Vier Jahre später entstand das erste arbeitsfähige Netz, welches die unterschiedlichen paketorientierten Netzwerke verband. Die ersten Teilnehmer des Internets waren das ARPANET, ein paketorientiertes Funknetzwerk, ein paketorientiertes Satellitennetzwerk und das von XEROX-PARC entwickelte Ethernet. Das Kommunikationsprotokoll, daß alle diese einzelnen Netzwerke verband, war das TCP/IP.

1983 hatte das ARPANET eine solche Ausdehnung erreicht, daß es nicht mehr weiter unter dem Status eines Forschungsnetzwerkes betrieben werden konnte. Die Kontrolle wurde deshalb von der DARPA an die **Defense Communication Agency (DCA)** übergeben. Gleichzeitig geschah die Umstellung aller ARPANET-Knoten auf das **Übertragungsprotokoll TCP/IP**. Weiters erfolgte eine Aufteilung des Netzes in einen **militärischen Teil (MILNET)** und in einen **forschungsorientierten Teil (ARPANET)**.

Am 24. Juni **1983** wurde das von XEROX und DEC entwickelte Verfahren zur Übertragung von Daten in lokalen Netzwerken (**Ethernet**) vom IEEE zum internationalen Standard 802.3 erhoben. Diese Ethernet-Technologie ermöglichte eine einfache Verbindung von Computersystemen zu einem lokalen Verbund. In weiterer

Folge wurde das Ethernet in Tausenden von kleinen Netzwerken verwendet, die ihrerseits wieder über das Internet miteinander verbunden wurden.

Neben dem ARPANET und dem MILNET entstanden in den 80er Jahren noch etliche Weitverkehrsnetze wie z.B. das **USENET**, das Netz der NASA, oder das **BITNET**. Auch die **NSF (National Science Foundation)** richtete ein unabhängiges, wissenschaftlich orientiertes Netzwerk ein.

Ein weiteres Projekt an dem die NSF zu Beginn der 80er Jahre beteiligt war, führte zu einer Erneuerung der Hauptübertragungswege (**Backbones**) des nordamerikanischen Internet. Ziel war es, die mit Supercomputern ausgestatteten Forschungszentren miteinander zu verbinden. Dadurch sollte einem größeren Kreis von Forschern der Zugang zu Hochleistungscomputern erleichtert werden. Aus diesem Grund wurde das **NFSNET** geschaffen, das zunächst mit einer Übertragungsrate von 56kbit/sek und später mit 1.5Mbit/sek arbeitete. Das NFSNET startete 1986 und löste die alten und langsamen Hauptverbindungsleitungen des Internet ab.

Der Name ARPANET hatte sich inzwischen von ARPA-Internet über Federal-Research-Internet und TCP/IP-Internet zu Internet gewandelt.

1987 beauftragte NSF das Unternehmen Merit-Network Inc. mit der Verwaltung und Wartung des Internet-Hauptnetzes. Bald erkannte man, daß die 1.5Mbit/sek Leitungen die sprunghaft anwachsende Netzwerkbelastung nicht mehr bewältigen konnten.

Zu Beginn der **90er Jahre** wurde das NSF-Backbone sukzessive auf Übertragungsgeschwindigkeiten von 45Mbit/sek umgestellt.

Seit **1993** wird an der Umstellung auf noch höhere Geschwindigkeiten gearbeitet. Die zugrundeliegende Technologie ist ATM (Asynchronous Transfer Mode) wobei Geschwindigkeiten von 155Mbit/sek bis zu 622Mbit/sek vorgesehen sind.

Zu **Beginn der 80er Jahre** konzentrierten sich die europäischen Internet-Aktivitäten vor allem auf die skandinavischen Länder. Norwegen, Schweden, Finnland und Dänemark waren im universitären Bereich schon viel enger an Nordamerika als an Europa angebunden. Deswegen fand eine rasche Verbreitung des Internet zu Beginn vor allem in Nordeuropa statt. Unter **NORDUnet** erreichte dieses Netz eine große Ausdehnung. In weiterer Folge entstand in den meisten europäischen Ländern nationale Forschungsnetze mit Verbindungen (Gateways) in das weltweite Internet. Bedeutende Netzwerke waren **EARN**, **EUnet** und der europäische Teil des nordamerikanischen **CSNET**. Ein gemeinsames europäisches TCP/IP-Netzwerk gab es jedoch erst zu Beginn der 90er Jahre.

Die **EBONE-Initiative** (European Backbone) der europäischen Internet-Organisation RARE (Réseaux Associés pour la Recherche Européenne) ermöglichte **1992** den europaweiten Internet-Betrieb. Im Rahmen des EBONE-Projektes wurden erstmals

zentrale europäische Internet-Hauptverkehrswege geschaffen. Vor dieser Zeit war es durchaus möglich, daß die Kommunikation zwischen 2 europäischen Netzen nicht direkt sondern über Nordamerika erfolgte.

1991 wurde das **NREN (National Research and Education Network)** im Rahmen des High Performance Computing Act (HPCA) des amerikanischen Kongresses als Nachfolger für das NFSNET bestimmt. Er legt Planung und Finanzierung eines Programmes zur Förderung der Forschung im Bereich Hochgeschwindigkeitsnetze fest. **1994** wurden dafür mehr als 1 Milliarde US-Dollar ausgegeben. Diese Ergebnisse sollen die Grundlage für die Realisierung einer neuen nationalen Informationsinfrastruktur (National Information Infrastructure - NNI) bilden. Bis zum Jahr 2007 werden laut amerikanischen Vizepräsidenten Al Gore mehr als 500 Milliarden US-Dollar für eine Verbesserung des nordamerikanischen Internets auf der Basis der HPCA-Technologien verwendet.

2.2 Organisationsstrukturen im Internet

Aufgrund der vielschichtigen Entwicklungen im Internet besteht auch eine organisatorische Infrastruktur aus einer Vielfalt von unterschiedlichsten nationalen und internationalen Organisationen und Vertretungen [Kyas 94].

2.2.1 Internet Society (ISOC)

1991 wurde von CNRI (Corporation for National Research Initiatives), EDUCOM, RARE und IAB die ISOC gegründet. Ziel der ISOC ist es, die internationale Weiterentwicklung des Internets zu koordinieren, sowie bestehende Dienste zu verbessern bzw. zu erweitern. In einer vierteljährlich erscheinenden Publikation (**Internet Society News**) wird über die Aktivitäten informiert. Einmal jährlich erfolgt eine Konferenz, die INET.

2.2.2 IAB, IETF und IRTF

Die technischen Entscheidungen und Richtlinien für das nordamerikanische Internet werden seit **1983** vom Internet Activities Board (IAB) getroffen. Im Jahre 1989 wurde das IAB reorganisiert. Die Aktivitäten wurden auf die beiden Hauptabteilungen Internet Reseach Task Force (IRTF) und Internet Engineering Task Force (IETF) aufgeteilt. Das IRTF ist für kurz- und mittelfristige Projekte zuständig, die IETF für die organisatorischen Bereiche. Die verschiedenen Aufgaben sind dabei jeweils in Arbeitsgruppen organisiert. **1991** wurde das IAB in die ISOC eingegliedert. Seitdem arbeiten in den Arbeitsgruppen von IRTF und IETF zunehmend auch internationale

Vertreter mit. Eine der wichtigsten Aufgaben der IETF ist es, Spezifikationen zu entwickeln, die in Zukunft zu Internet-Standards werden sollen.

2.2.3 RARE und RIPE

Die europäische Organisation RARE wurde **1986** mit dem Ziel gegründet, den Aufbau einer paneuropäischen Hochgeschwindigkeitsinfrastruktur für die Datenübertragung zwischen den Forschungs- und Entwicklungseinrichtungen der verschiedenen Ländern zu fördern. RARE ist einer der Mitbegründer der ISOC und aktives Mitglied im europäischen Standardisierungsgremium ETSI (European Telecommunications Standards Institute). RARE ist heute die Dachorganisation aller europaweiten Internetaktivitäten. Die ursprünglich von verschiedenen europäischen Internetbetreibern gegründete RIPE (Réseaux IP Européens) hat die Aufgabe, die unterschiedlichen, das TCP/IP betreffenden Aktivitäten in Europa zu koordinieren.

2.2.4 Schaffung von Internet-Standards

Vorschläge, Ideen, Richtlinien und Spezifikationen die das Internet betreffen, werden in Form von **RFCs (Requests for Comments)** veröffentlicht. Seit **1989** ist die Struktur und der Aufbau von RFCs durch die RFC 1111 geregelt. Ein RFC-Editor koordiniert als Mitglied des IAB die Veröffentlichung sowie die Verteilung im Internet. Die einzelnen RFCs werden in der Reihenfolge ihres Erscheinens sequenziell numeriert. RFC 1000 ordnet die RFCs 1 bis 999, die im Zeitraum 1969 bis 1987 erschienen sind, nach thematischen Gebieten. Die Verwaltung aller RFCs erfolgt durch das Informatik-Institut der Universität von Südkalifornien (USC-ISI).

Alle RFCs, die den Status von offiziellen IAB-Internet-Standards erreichen, werden in der RFC 1600 zusammengefaßt. Ein Dokument, welches sich zu einem Internet-Standard entwickeln soll, muß drei Stadien erfolgreich absolvieren:

Zunächst wird geprüft, ob die beschriebene Spezifikation ausreichend stabil ist, keine Probleme bekannt sind und der Vorschlag technisch ausgereift ist. Danach wird das Dokument zu einem Standardvorschlag (Standard-Proposal) ernannt. Nachdem entsprechend dem Standardvorschlag mindestens 2 unabhängige Implementationen im Internet realisiert sind, und diese beiden Implementationen auch zusammenarbeiten können, wird das Dokument zu einem Draft-Standard. Nach weiteren 4 Monaten kann, falls sich die Spezifikation bewährt hat und genügend Interesse innerhalb der Internet-Organisationen besteht, der Draft-Standard in einen Internet-Standard umgewandelt werden.

2.2.5 Internet Assigned Numbers Authority (IANA)

Bestimmte Dienste im Internet wie z.B. Telnet, email, ftp erfordern eine global einheitliche Konfiguration auf allen am Internet angeschlossenen Computern. Das Kommunikationsprotokoll TCP/IP legt aus diesem Grund für die wichtigsten Funktionen einheitliche Softwareadressierungen fest, unter denen global ein und dasselbe Funktionsmerkmal implementiert ist. Diese Adressen werden als **TCP/IP-Protokoll-Ports** oder kurz Ports bezeichnet. Ein Port ist eine logische Zieladresse innerhalb eines Netzwerkknotens, die, wird sie angesprochen, eine bestimmte Funktion ausführt. Den wichtigsten Diensten im Internet werden von einer zentralen Stelle, der IANA, Portnummern zugeordnet. Zum Beispiel dient die Portnummer 23 dem Telnet-Dienst und die Portnummer 21 dem ftp. RFC 1060 enthält eine Liste aller universell genutzter Portnummern.

2.2.6 Network Information Center (NIC)

Um sicherzustellen, daß jedes an das Internet angeschlossene Netzwerk unterschiedliche Adressen verwendet, werden die Internetadressen von einer zentralen Stelle vergeben, dem NIC. Das NIC vergibt dabei die eigentliche Internet-Adressvergabe an regionale Internet-Registaturen (Internet Registry). Für Europa ist das RIPE-NCC (RIPE Network Coordination Center) zuständig.

2.3 Übertragungsmechanismen

2.3.1 Kommunikationsprotokolle im Internet

Die Vorschriften nach denen die Kommunikation im Internet erfolgt, werden als Protokoll bezeichnet. Üblicherweise sind diese Protokolle in einzelne Module unterteilt, wobei jedes Modul eine bestimmte Aufgabe innerhalb des Kommunikationsprozesses hat (z.B. Verbindungsaufbau, Datenübertragung, Fehlerkontrolle,...).

Die Funktionen der einzelnen Module wurden 1983 von der International Standard Organisation (ISO) standardisiert und auf 7 Module festgelegt (**OSI 7-Schichten-Modell**).

OSI-Schicht 1: Bitübertragungsschicht (Physical Layer)

Die Bitübertragungsschicht legt die elektrischen und mechanischen Parameter einer Kommunikationsübertragung fest. Das Vorhandensein einer Schicht-1-Verbindung ist somit die Voraussetzung für eine Kommunikation.

OSI-Schicht 2: Sicherungsschicht (Data Link Layer)

Die Sicherungsschicht hat die Aufgabe, die zu versendenden Daten in Datenpakete aufzuteilen. Dafür werden bestimmte Bitmuster eingefügt, die das Erkennen von Beginn und Ende der Datenpakete ermöglichen. Werden Datenpakete nicht oder nur fehlerhaft übertragen, kümmert sich diese Schicht um eine erneute Übertragung der zerstörten oder verlorengegangenen Daten.

OSI-Schicht 3: Vermittlungsschicht (Network Layer)

Die wichtigste Aufgabe der Vermittlungsschicht ist die Auswahl der Übertragungswege für die Datenpakete. In dem weitverzweigten Netzwerk gibt es eine Vielzahl von möglichen Wegen vom Sender zum Empfänger. Die Vermittlungsschicht wählt nach verschiedenen Kriterien wie Übertragungszeit und Auslastung des Übertragungsweges den optimalen Weg für die Datenpakete aus.

OSI-Schicht 4: Transportschicht (Transport Layer)

Die Transportschicht übernimmt die Daten der ihr übergeordneten Sitzungsschicht, paßt sie an die Formate der OSI-Schicht 3 an und reicht sie an diese weiter.

OSI-Schicht 5: Kommunikationssteuerungsschicht (Session Layer)

Bei Multi-User-Systemen haben mehrere Benutzer gleichzeitig die Möglichkeit, den gleichen Computer zu benutzen. Die Aktivität jedes Benutzers wird dabei als Sitzung (session) bezeichnet. Die Kommunikation von Sessions mit der Außenwelt wird von dieser Schicht administriert.

OSI-Schicht 6: Darstellungsschicht (Presentation Layer)

Die Darstellungsschicht konvertiert die internen Datenformate von Computersystemen in die Standarddarstellung der Kommunikationsbeziehung, bzw. des Netzwerkes. Diese Schicht erspart dem Anwender die manuellen Einstellungen und Anpassungen des eigenen Systems an die Netzwerkstandardeinstellungen.

OSI-Schicht 7: Anwendungsschicht (Application Layer)

Die Anwendungsschicht beinhaltet die Dienstprogramme für die verschiedenen Funktionen, die über die Kommunikationsbeziehung, bzw. das Netzwerk realisiert werden sollen. In diesen Bereich fallen z.B. Terminalprogramme, email oder ftp.

2.3.2 TCP/IP

TELNET	FTP	SMTP	DNS	NTP	NFS
TCP			UDP		
IP					
Ethernet	ISO 8802-2		X.25	SLIP	PPP
(CSMA/CD)	(Token Ring)		ISO7776 (HDLC)	(seriell)	

Tabelle 1: Die Familie der TCP/IP-Protokolle

2.3.2.1 Internet Protocol (IP)

Das Internet-Protokoll ist das Basis-Kommunikationsprotokoll im Internet [Comer 88]. Die Daten werden paketorientiert übertragen. Damit legt IP das Paketformat aller Datenübertragungen im Internet fest. Die Pakete werden verbindungslos übertragen, d.h. daß jedes Paket für sich betrachtet wird und die Übertragung unabhängig von vorhergehenden oder nachfolgenden Paketen erfolgt. Allerdings ist kein Mechanismus vorgesehen um verlorene oder zerstörte Pakete erneut zu übertragen.

Die maximale Länge von IP-Paketen ist auf 65.535 Bytes beschränkt. Je nachdem, über welche Netzwerke ein IP-Paket übertragen wird, kann die Notwendigkeit bestehen, dieses Paket zu fragmentieren. In lokalen Netzwerken vom Typ Ethernet ist z.B. die maximale Paketlänge auf 1492 Bytes beschränkt. Auch manche Gateways sind nicht in der Lage, Pakete mit der maximalen Länge von 65.535 Bytes zu übertragen. IP spezifiziert deshalb, daß ein Internetgateway Pakete mit einer Mindestlänge von 576 Bytes verarbeiten können muß. Jedes IP-Fragment hat selbst wieder das Format eines gewöhnlichen IP-Paketes.

Ein wichtiges Feld innerhalb von IP-Paketen ist das TimeToLive (TTL) Feld. Dieses gibt die maximale Zeitdauer an, während der sich das Paket im Internet befinden darf. Während des Transportes des Paketes wird dieses Feld fortlaufend erniedrigt. Erreicht es den Wert 0, so wird das Paket verworfen.

2.3.2.2 Transmission Control Protocol (TCP)

Das Transmission Control Protocol ist das zweitwichtigste Protokoll im Internet [Comer 88]. Der wesentliche Unterschied zu IP ist, daß TCP die Daten im Rahmen einer virtuellen Verbindung garantiert überträgt. D.h. TCP hat Mechanismen, die

überprüfen, ob ein Datenpaket tatsächlich beim Empfänger eingetroffen ist oder nicht. Geht ein Paket verloren, oder wird es verändert, wird automatisch dieses Paket erneut angefordert.

Das TCP unterteilt den zu übertragenden Datenstrom zunächst in Segmente. Je nach Kapazität der Bufferspeicher der kommunizierenden Systeme wird dazu eine maximale Segmentgröße ausgehandelt. Die Standardgröße ist 675 Bytes minus den 40 Bytes für den IP-Header. Jedes Segment wird vor seiner Übertragung mit einer Nummer versehen. Der Empfänger kann durch Bezugnahme auf diese Nummer den Empfang bestätigen. Bei einer Fenstergröße von n kann die Sendestation n Pakete übertragen, ohne auf eine Empfangsbestätigung zu warten. Aber spätestens nach Versenden des n -ten Paketes muß einer Empfangsbestätigung für Paket Nr. 1 oder eins der nachfolgenden Pakete eingelangt sein. Erst dann darf Paket $n+1$ gesendet werden. Der Empfänger bestätigt immer ganze Gruppen von Paketen. Wird mit der ersten Empfangsbestätigung Paket $n-1$ bestätigt, so gelten damit auch alle vor Paket $n-1$ gesendeten Pakete als bestätigt.

Eine TCP-Verbindung kann aber nicht nur zwischen zwei Computern mit entsprechenden Internet-Adressen bestehen. Soll z.B. mit einem Computer, der ein multitaskingfähiges Betriebssystem besitzt, eine Verbindung aufgebaut werden, so muß vor dem Aufbau der Verbindung zusätzlich der sogenannte TCP-Port bekannt sein.

Diese Ports sind Softwareadressen, an denen sich die auf TCP aufsetzenden Dienstprogramme orientieren. Multitaskingfähige Computer sind so in der Lage, mehrere Kommunikationsprozesse gleichzeitig ablaufen zu lassen, wobei jedem Prozeß eine Portnummer zugewiesen wird. Einige Portadressen sind internetweit für bestimmte Dienstprogramme reserviert, andere Ports können dynamisch während des Verbindungsaufbaues zugeordnet werden.

2.3.3 Internet-Adressen

Die im Internet benutzte Adressierungsstruktur wurde mit dem Ziel geschaffen, eine möglichst effiziente Übertragung der Datenpakete zu gewährleisten. Die Internet-Adressen geben nicht nur Auskunft, wo der betreffende Netzteilnehmer ist, sondern darüber hinaus auch, über welche Übertragungswege die Daten transportiert werden. Jedem Computer am Internet wird eine, weltweit eindeutige 32bit (4Byte) große Zahl zugeordnet. Jede dieser Adressen setzt sich aus 2 Teilen zusammen:

- Netzwerkidentifikation (netid)
- Benutzeridentifikation (hostid)

Es gibt **5 verschiedene Klassen** von Internetadressen. Die Klassen A, B und C unterscheiden sich durch die Länge der Netzwerkidentifikationsnummer.

Die Schreibweise von Internet-Adressen ist üblicherweise in der Form, daß jedes Byte durch eine Dezimalzahl dargestellt wird und durch einen Punkt von den anderen getrennt wird (z.B.: 127.29.181.74). Jedem an das Internet angeschlossene Netzwerk wird so eine eindeutige Identifikation zugeordnet. Die zu der jeweiligen Netzidentifikation gehörende Benutzeridentifikation kann der lokale Netzwerkbetreiber selbst vergeben.

Für **Klasse A** Netzwerken stehen 254^3 Benutzeridentifikationen zur Verfügung. Allerdings gibt es weltweit nur 126 Klasse A Netzwerke.

Die 16.383 **Klasse B** Netzwerke können jeweils 254^2 Benutzer beinhalten, die 2.097.151 **Klasse C** Netzwerke je 254 Benutzeridentifikationen. Die Zahl der zu vergebenden Internetadressen ist also beschränkt, und aufgrund des starken Wachstums des Internet werden unter Umständen in absehbarer Zeit die Internetadressen verbraucht sein. Aus diesem Grund wurde im November 1991 vom IAB (Internet Activities Board) die ROAD-Gruppe (Routing and Adressing) gegründet. Die Aufgabe dieser Gruppe ist es, ein neues Adressierungs- und Routingkonzept zu erarbeiten.

Die Internet-Adressen der **Klasse D** werden als Multicastadressen bezeichnet. Mit Hilfe von diesen Adressen können Datenpakete an bestimmte Gruppen von Empfängern gesendet werden. Jeder Multicastgruppe ist eine bestimmte Klasse D Adresse zugeordnet. Bestimmte Multicastadressen werden dabei wie alle anderen Internet-Adressen zentral vergeben, andere sind zur vorübergehenden Benutzung frei verfügbar. Eine Multicastadresse kann aber immer nur als Zieladresse genutzt werden und nicht als Sendeadresse.

Die **Klasse E** ist für zukünftige Anwendungen reserviert und wird zur Zeit nicht verwendet.

2.3.3.1 Besondere Internet-Adressen

Einige Netzwerkadressen sind für spezielle Funktionen reserviert. Die beiden wichtigsten sind die Broadcast- und Loopback-Adresse.

Eine **Broadcast-Adresse** dient dazu, daß ein bestimmtes Datenpaket an alle Knoten eines Netzwerkes verschickt wird. Alle Bits des Feldes für die Benutzeridentifikation werden auf 1 gesetzt. So ist z.B. die Broadcast-Adresse für das Netzwerk 121 der Klasse A 121.255.255.255

Die Klasse A Adresse 127 ist für **Loopback** reserviert. Ein Paket mit der Zieladresse 127.0.0.0 wird deshalb, ohne jemals wirklich in das Internet zu gelangen, unmittelbar wieder beim Sender eintreffen. Diese Loopback-Funktion kann zum Überprüfen von Netzwerksoftware verwendet werden.

Wird als Netzwerkidentifikation 0 angegeben, so bezeichnet dies automatisch das eigene lokale Netzwerk. Die Benutzeridentifikation mit dem Wert 0 entspricht dem eigenen Netzwerkknoten.

2.3.3.2 IPv6 (Internet Protocol version 6)

Das IPv6 ist ein neues System, welches zur Zeit entwickelt wird, um die Adressierung von Computern im Internet zu verbessern. Die Internet Engineering Task Force (IETF) hat beschlossen, daß dieses Protokoll die nächste Generation für die IP-Adressierung werden soll. Dadurch wird diese Version 6 die bisher bestehende Version 4 ersetzen.

In der Version 4 besteht eine IP-Adresse aus 4 8-bit Zahlen, die durch einen Punkt voneinander getrennt sind.

z.B.: 128.53.24.207

In diesem Schema sind mehr als 4 Milliarden Adressen möglich. Die Hierarchie dieser Adressen ist in 2 Ebenen aufgeteilt. Die Adresse besteht aus einem Netzwerkteil und einen Host in diesem Netzwerk. Die Praxis hat aber gezeigt, daß dies eine sehr ineffiziente Methode ist, um den IP-Adressraum aufzuteilen.

In der neuen Version 6 werden 128-bit IP-Adressen vergeben. Jede Adresse besteht aus 8 Teilen, wovon jeder eine 16-bit Zahl enthält.

z.B.: 1023:0:24:A8:34:10A3:34E:F03B

Der Vorteil von IPv6 ist nicht nur der wesentlich größere Adressraum, sondern auch die flexible Architektur. Zusätzlich ist es noch voll kompatibel zur bestehenden Version 4.

2.3.4 Domain Name System (DNS)

Für Menschen ist es schwierig sich viele einzelne Internet-Adressen zu merken, da es immer nur 32bit große Zahlen sind. Es ist wesentlich einfacher, mit Namen anstatt mit Zahlen zu arbeiten. Bis in die erste Hälfte der 80er Jahre wurde jedem Computer einfach neben seiner numerischen Adresse ein Name zugeordnet. 1986 enthielt die offizielle Namensliste des Internet ca. 3600 Einträge. Diese flache Namensstruktur wurde aber im Zuge des raschen Wachstums des Internets sehr schnell unbrauchbar. Aus diesem Grund wurde **1986** mit dem **Domain Name System** ein hierarchisches Namenssystem eingeführt. Die einer anderen Domain hierarchisch untergeordnete Domain wird als Sub-Domain bezeichnet. Der Domain-Name besteht aus einer **Top-Level-Domain** (der rechteste Teil des Namens), sowie aus **Subdomains**. Die Domain-Namen sind unabhängig davon, ob sie in Groß- oder Kleinbuchstaben geschrieben werden.

Top Domains	
COM	Kommerziell
EDU	Erziehung und Ausbildung
MIL	Militär
ORG	Organisationen
GOV	Regierungsstellen
NET	Netzwerkbetreiber

Tabelle 2: Top Domains im Internet

Zusätzlich gibt es noch für jeden Staat eine Kurzbezeichnung als Top-Level-Domain.

Zukünftig sollen die Top Domains um weitere Elemente erweitert werden:

Zukünftige Top Domains	
ARTS	Kunst
REC	Freizeit, Erholung
NOM	Privatpersonen
STORE	Firmen
INFO	Informationen
FIRM	Firmen
WEB	Provider

Tabelle 3: zukünftige Top Domains im Internet

2.4 Dienste

Im Internet werden verschiedenste Dienste angeboten, die zur Kommunikation (synchron oder asynchron) und für den Transport von Daten verwendet werden. Ihnen allen ist gemeinsam, daß sie weltweit genau definiert sind und daher im gesamten Internet verwendet werden können. Das Basiskommunikationsprotokoll ist immer das Internet-Protokoll (IP). Auf dieses wird dann meist TCP oder UDP aufgesetzt [Krol 94].

2.4.1 Email

Email ist das elektronische Äquivalent zum Versenden eines Briefes mit der Post. Um emails verschicken oder empfangen zu können, benötigt man einen Computer, ein Modem, eine gültige email-Adresse und ein email-Programm. Mit email ist es sehr einfach an eine einzelne Person oder an eine bestimmte Gruppe von Personen eine Nachricht zu schicken. Weiters besteht auch die Möglichkeit, zusätzlich zum eigentlichen Text noch Dokumente, Bilder, Tabellenkalkulationen oder Programme mitzuschicken.

Email-Programme bieten meist folgende Funktionen:

Cc: Eine email wird gleichzeitig an mehrere Personen verschickt. Jeder Empfänger erhält zusätzlich eine Liste, an wen diese email noch verschickt worden ist.

Bcc: Eine email wird gleichzeitig an mehrere Personen verschickt. Diesmal kann jeder Empfänger aber nicht erkennen wer, außer ihm, diese email noch erhalten hat.

Reply: Mit Hilfe dieser Funktion ist es sehr einfach auf eine erhaltene email zu antworten, bzw. die erhaltene email zu kommentieren und an den Absender zurückzuschicken. Mit Reply lassen sich sehr einfach Diskussionen führen.

Forward: Eine erhaltene email wird an eine weitere Person weitergeleitet. Der Absender hat zusätzlich noch die Möglichkeit, die weiterzuleitende email zu kommentieren.

Attachment: Zusätzlich zur email können noch beliebige Dateien mitgeschickt werden.

Nicknames: Die meisten email-Programme bieten die Möglichkeit, oft genutzte email-Adressen in einer eigenen Adressverwaltung zu speichern und wieder schnell abzurufen.

Empfangs- und Lesebestätigung: Manche Systeme erlauben die Möglichkeit, daß der Sender der email eine Bestätigung erhält, daß seine email beim Empfänger ordnungsgemäß angekommen ist, oder daß der Empfänger diese email gerade aufgerufen hat.

Gängige Email-Protokolle sind SMTP, POP3 und IMAP.

2.4.2 ftp (file transfer protocol)

Mit ftp kann man beliebige Programme oder Dateien, die auf einem speziellen Computer, einem ftp-Server im Internet angeboten werden, herunterladen (download). Ebenso ist auch der umgekehrte Weg möglich, daß Dateien auf diese ftp-Server geladen werden können.

2.4.3 telnet (remote login)

Telnet erlaubt es, von einem Computer aus auf ein, unter Umständen geographisch weit entferntes Computersystem zuzugreifen. Es besteht dann die Möglichkeit auf diesem Computersystem genauso zu arbeiten, als ob man dort physikalisch anwesend wäre.

2.4.4 Usenet News

Eine Art elektronisches Schwarzes Brett oder elektronisches Diskussionsforum. Jeder Benutzer kann beliebige Beiträge schreiben, diese werden automatisch über das Internet verbreitet und jede weitere Person, die mit news arbeitet, kann dann diesen Artikel lesen. Aufgrund der sehr hohen Anzahl der Benutzer sind die Artikel in verschiedenste Diskussionsforen aufgeteilt.

2.4.5 World Wide Web (WWW)

2.4.5.1 Geschichtliche Entwicklung

1989 begannen Überlegungen von Tim Berners-Lee und Robert Cailliau über die Verwendung eines Hypertextsystemes für die Vereinfachung von internationaler Zusammenarbeit. Sie bauten auf einer Idee von Ted Nelson aus dem Jahre 1965 auf. Das Problem, das sie lösen wollten, war folgendes: Forscher im Bereich Hochenergiephysik bei CERN wollten Informationen untereinander austauschen. Alle bisherigen Versuche sind aber fehlgeschlagen, da es sehr viele verschiedene Netzwerke und auch verschiedenste Übertragungsprotokolle gab. Ein weiteres Problem waren die unterschiedlichen Computertypen, die die Forscher benutzen.

Der erste technische Durchbruch gelang Nicola Pellow, einem Studenten bei CERN. Er entwickelte einen zeichenorientierten Browser im Jahr **1991**. Ein Jahr später war auch die Einbindung anderer Internetprotokolle (Gopher, Telnet, ftp) möglich.

1993 entwickelte das National Center for Supercomputing (NCSA) den ersten graphischen Browser (MOSAIC). Mit diesem Browser war es erstmals möglich, verschiedene Schriften und auch Bilder innerhalb des Hypertextes darzustellen. Damit war das WWW das erste System, mit dem es auf einfache Weise möglich war, optisch

ansprechende Information anzubieten. Dies war der Grundstein für den großen Erfolg und die weite Verbreitung des WWW.

2.4.5.2 Konzept des WWW

WWW ist in der offiziellen Beschreibung ein Internet-wide distribution hypermedia informationretrieval system (weltweites verteiltes Hypermedia Informationsabfragesystem) welches Zugriff auf eine sehr große Datenmenge bietet. WWW benutzt, wie viele andere Applikationen im Internet, das Client-Server-Modell. Die Clients sind nicht mehr als eine Sammlung von Programmen welche Anfragen an einen Server schicken können und die Antworten des Servers dann am Bildschirm visualisieren. Der Server ist ein Programm, das auf die Anfragen der Clients wartet und dann entsprechende Information an die Clients zurückliefert.

Damit dieser Informationsfluß möglich wird, wurden **3 Standards** definiert: Der **Uniform Resource Locator (URL)** dient als ein konsistentes und kompaktes Adressierungsschema. Das **Hypertext Transfer Protocol (HTTP)** schafft die Voraussetzungen für die Datenübertragung und die **Hypertext Markup Language (HTML)** ist das Standard-Format bei der Gestaltung von Hypertexten.

2.4.5.3 HyperText Markup Language (HTML)

Der Grundstein für die Entwicklung von Hypertext wurde **1945** von Vannevar Bush gelegt. In seinem Artikel "As We May Think" im "The Atlantic Monthly" prognostizierte er, daß der Fortschritt der Wissenschaft stagnieren wird, weil die Forscher nicht mehr in der Lage sein werden, relevante Informationen schnell zu finden. Um diesem Problem entgegenzuwirken stellte er sein System Memex vor. Dieses System soll in der Lage sein, Dokumente zu verwalten. Die Verbindung zwischen den Dokumenten soll durch Links erfolgen. Weiters soll es auch die Möglichkeit sogenannter trails geben, das sind vordefinierte Wege durch eine Auswahl von Dokumenten zu einem bestimmten Thema. Wegen der damals begrenzten technischen Möglichkeiten konnte Memex nicht in der Praxis erprobt werden. Trotzdem hat Bush schon damals auf einige wichtige Funktionen hingewiesen, die heute noch immer ihre Notwendigkeit haben: schneller Zugriff auf Informationen, die Möglichkeit Dokumente zu annotieren, Dokumente miteinander zu verlinken und die Links zu speichern.

1965 stellte Theodor Holme Nelson erstmals den Begriff **Hypertext** vor. Sein Ziel war es, mit dem Projekt Xanadu ein System zu schaffen, mit dem die gesamten Publikationen der Menschheit gespeichert werden können. Jeder Zugriff auf ein Dokument soll dem Nutzer einen sehr kleinen Geldbetrag kosten, der dann dem Autor des Dokumentes zur Verfügung gestellt wird.

1983 wurde **Intermedia** entwickelt. Es war ein auf Hypertext basierendes Informationssystem, welches Links in jedem Dokumenttyp erlaubte, auch in Bildern, Filmen oder Audiodokumenten.

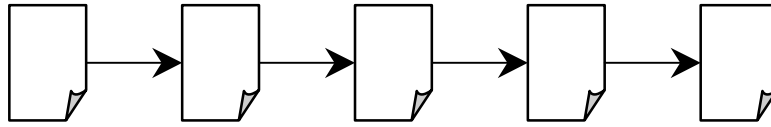


Bild 2: Linearer Text

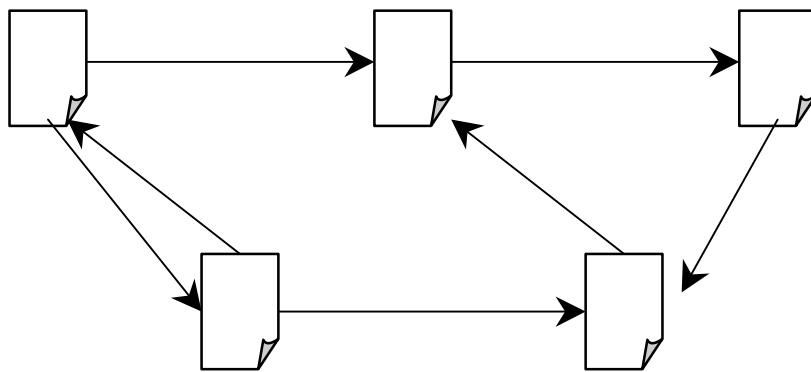


Bild 3: Hypertext

Ein linearer Text, z.B. ein Buch, wird sequentiell gelesen. Ein Hypertext besteht aus vielen kleinen Informationseinheiten, die durch Links miteinander verbunden sind. Es ist kein Pfad vorgegeben, in welcher Reihenfolge die Informationen betrachtet werden müssen. Der Anwender kann sich jene Informationen herausuchen, die ihn interessieren, die restlichen Informationen zieht er in seine Betrachtung nicht ein [Berk 91].

HTML ist das wichtigste Dokumentformat im WWW. Es ist eine Anwendung von **SGML (Structured Generalized Markup Language)**. Das HTML-Dokument wird in verschiedene Teile zerlegt, einen Kopf, Paragraphen, Fußnoten, usw. Es ist das Ziel einer Markup Language, die logische Struktur eines Dokumentes zu beschreiben und nicht das aktuelle Layout an einem bestimmten Bildschirm. Mit dieser Eigenschaft ist es auch möglich, HTML-Dokumente auf den verschiedensten Computersystemen darzustellen.

Ein HTML-Dokument besteht aus dem eigentlichen Text und einer speziellen Zusatzinformation (Markup) der die logische Struktur des Textes beschreibt. Alle

Textstellen sind von Markups eingeschlossen, die den Typ der Textstelle beschreiben. Die Markups selbst sind, um sie vom eigentlichen Text zu unterscheiden, in spitze Klammern eingeschlossen.

Ein Beispiel für ein einfaches HTML-Dokument:

```
<HTML>

<HEAD>

<TITLE>Dies ist der Titel</TITLE>

</HEAD>

<BODY>

<H1>Überschrift</H1>

Hier befindet sich der eigentliche Textkörper.

</BODY>

</HTML>
```

2.4.5.4 Hypertext Transfer Protocol (HTTP)

Der Sinn dieses Protokolles ist es, einen schnellen und flexiblen Mechanismus zur Verfügung zu stellen, um Referenzen zwischen Informationseinheiten zu verfolgen. Es dient nicht nur der Übertragung von Hypertexten, sondern auch jeder Art von Information, die bei der Verfolgung von Links benötigt wird. Das Protokoll ist stateless und objektorientiert. Das bedeutet, zwischen einem Client und einem Server wird keine Verbindung aufrechterhalten. Dies hat den Nachteil, daß bei Anfrage an den Server eine neue Verbindung aufgebaut werden muß.

2.4.5.5 Common Gateway Interface (CGI)

CGI ist ein Standard für das Interface zwischen externen Applikationen und einem Server. Ein HTML-Dokument, welches vom Server an den Client geschickt wird, ist statisch. Das bedeutet der Inhalt dieses Dokumentes ist immer gleich. Ein CGI-Programm wird aber in Echtzeit ausgeführt, so daß sein Output als dynamisches HTML-Dokument vom Server an den Client übertragen werden kann. Die Programmiersprache eines CGI-Programmes hängt nur davon ab, welche Sprachen am Computer des Servers möglich bzw. installiert sind. Die einzigen 2 Einschränkungen für CGI-Programme sind, daß sie nicht zu langsam sind, und daß sie nach außen abgesichert sind, damit Hacker keinen Zugriff auf den Server erhalten können.

3 HYPER-G (HYPERWAVE)

3.1 Die Entstehung von Hyperwave

Die Entwicklung von Hyperwave begann **1991**. Hyperwave baut auf den Ideen des WWW auf, verbessert jedoch einige Konzepte, die im WWW vernachlässigt worden sind. Hyperwave speichert Links nicht im Dokument selbst, sondern in einer eigenen Datenbank. Hyperwave unterstützt bidirektionale Links und auch automatische Link Konsistenz: Links auf nicht mehr vorhandene Seiten werden automatisch gelöscht, auch über Servergrenzen hinweg. Hyperwave besteht aus einer orthogonalen hierarchischen Struktur, integriertem Indizieren und verschiedenen Suchmöglichkeiten.

Im Juni **1994** wurde der erste offizielle Hyperwave Server vorgestellt, zusammen mit den Browsern Harmony (für UNIX) und Amadeus (für Windows). Da sich Hyperwave von einem Universitätsprototypen immer mehr zu einem universellen Produkt entwickelte, wurde im Herbst **1995** die Firma **Hyper-G R+D** in Graz gegründet. Allmählich stellten sich auch die ersten größeren Verkaufserfolge ein, wie z.B. der erste Auftrag von Sumitomo Electric Co. in Japan, die auch den Vertrieb von Hyperwave in Japan übernahm. Ein weiterer Großauftrag folgte von Motorola und in Korea übernahm die Softwarefirma KIES den Vertrieb von Hyperwave. Im Dezember übersiedelten zwanzig Hyperwave Mitarbeiter von der TU-Graz in die neue Firma Hyperwave Ges.m.b.H in Graz. Diese Firma ist eine Tochtergesellschaft der Hyperwave Ges.m.b.H. in München, die für das Marketing, den Vertrieb, den Support und die Schulungen verantwortlich ist.

1997 gab es eine kleine Sensation: das Produkt erhielt auf der **CeBit'97** den "**Best of the Show Award**" von BYTE. Etwas später kam es dann zur offiziellen Eröffnung der Hyperwave R+D Ges.m.b.H. durch den österreichischen Bundeskanzler Viktor Klima. Mit dabei waren auch der Rektor der Technischen Universität Graz Dr. I.Killman und die Landesräte Ing. Ressler und Dipl.Ing. Paierl.

3.2 Konzept von Hyperwave

3.2.1 Client-Server Architektur

Die meisten Informationssysteme benutzen eine **einfache Client-Server-Architektur** [Dalitz 95], [Dalitz 96], [Flohr 95], [Flohr 96], [MaurerH 96], [Sengstack 96]. Diese Architektur bedeutet, daß mehrere, voneinander völlig unabhängige Computer (Clients) über Datenleitungen, oder auch das Internet Dienste eines anderen Computers (Server) in Anspruch nehmen können. Dies können Informationsangebote, oder auch Rechenleistung sein. In Hyperwave wird im Gegensatz dazu eine **proxy Client-Server Architektur** verwendet. Dabei wird jeder Client mit nur einem Server verbunden. Dieser Server ist jener mit der besten Verbindung zum Client. Solche Server werden als **local server** oder als **proxy server** bezeichnet. Diese Architektur hat gegenüber der einfachen Client-Server-Architektur einige große Vorteile:

- Wenn der Server von einem anderen Server Informationen abfragt, so werden diese in seinem lokalen Cache zwischengespeichert. Wird dann dieses Dokument mehrmals angefordert, so muß es nicht mehr vom anderen Server geholt werden, sondern nur noch aus dem lokalen Cache. Dies führt zu einer beachtlichen Verminderung des Datentransports auf den Leitungen und damit auch zu einer kürzeren Ladezeit.
- Die Clients kommunizieren immer nur mit einem Server. Dadurch braucht immer nur die Verbindung zum Server aufgebaut und gehalten werden.
- Der Server kann auch die Aufgabe der Konvertierung von Dateiformaten oder Protokollen übernehmen. Diese Funktionalität muß dadurch nicht mehr vom Client wahrgenommen werden, was zu einer Vereinfachung der Clients führt.
- Für Informationsanbieter besteht die Möglichkeit den Datenverkehr der jeweiligen Clients zu überwachen, und dadurch auch eine etwaige Verrechnung vorzunehmen.

3.2.2 Kooperierende Prozesse

Der Hyperwave-Server besteht aus 3 miteinander kooperierenden Prozessen:

- **Document Server:** Er stellt den Clients lokale Dokumente zur Verfügung, übernimmt Dokumente vom Client und speichert sie. Falls von den Clients Dokumente angefordert werden, die auf anderen Servern liegen, werden sie geholt, den Clients zur Verfügung gestellt und zusätzlich in einem Cache für weitere Anfragen zur Verfügung gehalten. Zusätzlich werden noch CGI-Scripts gestartet und deren Ergebnisse an die Clients zurückgeliefert.

- **Link Server (Object Server):** Dies ist eine objektorientierte Datenbank, die Objekte und ihre Verbindung untereinander speichert und verwaltet. Mögliche Objekte in Hyperwave sind: Dokumente, Collections, Anker, User, Usergruppen, usw. Der ursprüngliche Name Link Server liegt daran, daß hier auch die vollständigen Informationen über die Hyperlinks gespeichert sind. Eine weitere Aufgabe dieses Prozesses ist es, jedem Objekt eine weltweit eindeutige ObjektID zuzuordnen, unter der dieses Objekt angesprochen werden kann. Beim Einfügen eines Objektes werden die Links von diesem Objekt getrennt und als eigenständige Objekte in der Datenbank abgelegt. Zusätzlich werden hier noch die Attribute (Metainformation) eines Objektes und auch seine Zugriffsrechte verwaltet.
- **Full Text Server:** Dieser Prozeß verwaltet den inverted index aller Dokumente am Server, um bei einer Suche schnell Ergebnisse liefern zu können. Jedes Textdokument, daß im Server gespeichert wird, wird zusätzlich in den Index aufgenommen.

3.2.3 Datenmodell

Hyperwave basiert auf einer objektorientierten Datenbank, die speziell für die Anforderungen eines verteilten Informationssystems entwickelt wurden. Die Daten in der Datenbank sind Objekte und ihre Beziehungen untereinander. Jedes dieser Objekte besitzt zusätzlich noch Metainformationen, wie z.B. Erstellungsdatum, Autor, Zugriffsrechte,...

Die wichtigsten Objekte in Hyperwave sind:

- **Document:** von diesem Basisobjekt sind viele weitere Typen abgeleitet:
 - **Text:** HTML-Dokumente, Plain-Text-Dokumente
 - **Image:** in den Formaten GIF, JPEG, TIFF, PNG,...
 - **Movie:** Filmsequenzen z.B. AVI, MPEG, MOV,...
 - **Sound:** Audiodaten z.B. AU, WAV, AIFF, MIDI,...
 - **3D-Scene:** VRML-Objekte
 - **Postscript:** Dokumente im Postscript-Format
 - **Remote Object:** Verbindungen zu weiteren Internet-Diensten: WWW, Gopher, WAIS, FTP, Telnet,...
 - **Generic Object:** benutzerdefinierte Objekte, z.B. Tabellen in Excel, Winword-Dokumente, Autocad-Zeichnungen, PowerPoint-Präsentationen,...
- **Collection:** Entspricht einem Directory (Folder) auf einem Dateisystem, ist aber wesentlich mächtiger. Dadurch läßt sich ein Informationsangebot sehr leicht

strukturieren, bzw. ein bestehendes sehr leicht warten und modifizieren. Es besteht die Möglichkeit alle Informationen auf einem Server durch Collections in eine baumartige Struktur zu bringen, die automatisch eine leicht verständliche und einfache Navigation durch das Informationsangebot zuläßt.

- **Cluster:** Ein Cluster ist eine spezielle Collection, die Dokumente, die in irgendeiner Form zusammengehören, zu einer Einheit zusammenfaßt. Es gibt unterschiedliche Arten von Clustern in Hyperwave:
 - **Language Cluster:** Implementation der Multilingualität von Hyperwave: Dokumente in verschiedener Sprache werden in diesem Cluster zusammengefaßt. Je nach ausgewählter Sprache wird ein Dokument entsprechend dieser Sprache dargestellt.
 - **Multimedia Cluster:** Mehrere Objekte, die die unterschiedlichen menschlichen Sinnesorgane ansprechen werden zu einer Einheit zusammengefaßt: z.B. ein Videofilm, ein Audiodokument und dazu passende Untertitel als Text.
 - **Multicluster:** Dieser Cluster hat die Eigenschaft, daß alle Dokumente, die sich darin befinden, sofort beim Betreten des Clusters gleichzeitig, untereinander erscheinen.
 - **Alternative Cluster:** Es besteht die Möglichkeit, daß ein Objekt in unterschiedlicher Qualität in diesem Cluster gespeichert wird. Je nach Einstellung des Benutzers (z.B. abhängig von seiner möglichen Übertragungsbandbreite) wird dann das für ihn passende Objekt dargestellt. Eine weitere Anwendungsmöglichkeit ist es, Dokumente in verschiedenen Formaten anzubieten, z.B. in HTML, Postscript oder Winword. Je nach Benutzereinstellung wird das passende Format ausgewählt.
- **Sequence:** In eine Sequence können beliebige Objekte eingespielt werden. Wird diese Sequence von einem Benutzer betreten, erhält dieser zunächst eine Übersicht über alle Dokumente, dann kann er diese Dokumente in sequenzieller Reihenfolge nacheinander betrachten. Die Navigation durch diese Liste von Dokumenten wird automatisch von Hyperwave generiert, sodaß es keine Probleme bereitet, Dokumente hinzuzufügen oder zu löschen.

3.2.4 Links in Hyperwave

Die Links in Dokumenten werden getrennt von den Dokumenten im Link Server gespeichert. Dieses Prinzip bietet eine große Zahl von Vorteilen gegenüber der herkömmlichen Speicherung von Links innerhalb von Dokumenten:

- Links sind bidirektional. Dadurch besteht die Möglichkeit, Informationen anzubieten, die angeben, welche Dokumente auf ein bestimmtes Dokument zeigen, bzw. welche Dokumente durch ein bestimmtes Dokument referenziert werden. Weiters lassen sich dadurch Navigationshilfen wie z.B. eine Local Map realisieren.
- Durch diese Bidirektionalität läßt sich Link-Konsistenz gewährleisten. Wenn beispielsweise ein Dokument gelöscht wird, dann werden auch alle Verweise zu diesem Dokument unsichtbar.
- In Hyperwave werden Links als Objekte realisiert. Dadurch können sie auch Metainformationen wie z.B. Zugriffsrechte, oder Erstellungsdatum beinhalten.
- Dadurch, daß Links unabhängig vom Dokument gespeichert werden, können sie in beliebigen Dokumenttypen aufscheinen, z.B. auch in Audio- oder Videodokumenten.
- Durch Erzeugen oder Verändern von Links, wird das Dokument nicht beeinflusst. Dadurch können auch Links in Dokumenten erstellt werden, auf denen man nur Lese- und keine Schreibrechte hat (z.B. bei einer CD-ROM).

In Hyperwave besteht ein Link immer aus einem **Source Anchor**, der beim Dokument, von dem er ausgeht definiert wird, und einem **Destination Anchor**, der als Ziel ein anderes Dokument, oder einen bestimmten Teil eines Dokumentes hat.

3.2.5 Suchmöglichkeiten in Hyperwave

In Hyperwave kann die Suche in zwei Arten unterteilt werden:

Suchbereiche: Dienen zur Einschränkung des Bereiches, in dem gesucht werden soll.

In Hyperwave existieren die folgenden Möglichkeiten, die Suche auf einen bestimmten Bereich einzuschränken:

- **Activated Collections:** Die Suche erfolgt nur in jenen Collections, die vom Benutzer für die Suche ausgewählt worden sind.
- **Whole home Database:** Die Suche erfolgt am gesamten lokalen Server.

Sucharten:

- **Objekttitel:** Es wird nach regulären Ausdrücken im Titel eines Objektes gesucht.
- **Objekteigenschaften:** Spezielle Objekteigenschaften, die bei einer Suchabfrage festgelegt werden können, sind z.B. der Autor, das Erzeugungsdatum oder Schlüsselwörter.
- **Volltextsuche:** Alle Textdokumente werden bei der Aufnahme in den Hyperwave-Server indiziert. Bei der Volltextsuche, wird nach den Suchbegriffen innerhalb der Textdokumente gesucht.

3.2.6 Benutzerverwaltung und Zugriffsrechte

Damit unter Hyperwave große Datenmengen von vielen Personen gleichzeitig gewartet werden können, müssen die Benutzer effizient verwaltet werden. Jedes Dokument kann durch das Rechtfeld in den Attributen so definiert werden, daß eine bestimmte Benutzergruppe oder auch einzelne Benutzer bei diesem Dokument Schreib- oder auch nur Leserechte haben.

Zur Identifikation von Benutzern gibt es 4 verschiedene Arten:

- **Identified Mode:** Die Benutzer müssen sich selbst mit Namen und Paßwort am Server identifizieren. Dadurch können sie Dokumente am Server erzeugen, modifizieren oder löschen.
- **Semi-Identified Mode:** Die Benutzer haben ebenfalls Schreibrechte. Aber nur das System selbst weiß, wie der Name des Benutzers wirklich lautet. Diese Identifikationsart kann sehr gut für Gruppendiskussionen am Server verwendet werden. Die einzelnen Gruppenmitglieder untereinander sind anonym. Wenn es Probleme gibt, kann aber der Systemadministrator den Verursacher herausfinden.
- **Anonymously Identified Mode:** Dieser Modus ist ähnlich dem Semi-Identified Mode, nur mit dem Unterschied, daß der wirkliche Name des Benutzers dem System nicht bekannt ist. Trotzdem haben Benutzer eine eigene Homecollection und können dort ihre Einstellungen und Links speichern.
- **Anonymous Mode:** Der Benutzer ist nicht identifiziert und hat daher aus Sicherheitsgründen keine Schreibrechte am Server. Weiters hat er auch nur begrenzte Leserechte. Er kann nur Dokumente betrachten, die öffentlich freigegeben wurden.

Der anonymous identified mode ist zur Zeit noch nicht implementiert.

3.2.7 Meta-Information (Attribute)

Jedes Objekt am Hyperwave-Server besteht zusätzlich noch aus einer Reihe von Attributen, die dieses Objekt beschreiben. Jedes Attribut besteht aus 2 Teilen: Dem Namen des Attributes und seinem Wert. Jedes Attribut hat unterschiedliche Aufgaben, einige dienen zur Identifizierung des Objektes, andere kontrollieren die Darstellung und weitere speichern den Autor und das Erstellungs- bzw. Modifikationsdatum. Ein eigenes Attribut ist für die Zugriffsrechte auf das Objekt vorgesehen. In diesem Attribut ist festgelegt, wer das Objekt lesen, modifizieren oder löschen darf. Benutzer mit Schreibrechten für ein bestimmtes Objekt dürfen die Attribute desselben modifizieren, bzw. neue einfügen oder bestehende löschen.

3.2.8 Mehrsprachigkeit

In Hyperwave werden Dokumente in mehreren Sprachen automatisch unterstützt. Der jeweilige Benutzer legt seine Sprache fest, und automatisch erscheinen die Dokumente in der gewünschten Sprache. Falls das Dokument in der gewünschten Sprache nicht verfügbar ist, wird die nächstpassende Sprache dargestellt.

Von Hyperwave werden zur Zeit folgende Sprachen unterstützt:

- Deutsch
- Englisch
- Französisch
- Italienisch
- Spanisch
- Japanisch
- Koreanisch
- Ungarisch

Es ist aber möglich, daß beliebige Sprachen von Hyperwave unterstützt werden.

3.3 Wavemaster

Der Wavemater ist das Gateway zwischen einem Hyperwave-Server und einem WWW-Browser. Er kontrolliert den Aufbau und die Darstellung der Information am Browser. Zusätzlich bietet er noch die Möglichkeit, daß Daten direkt über diesen Browser am Server gewartet werden können.

In der Kopfzeile jedes Dokumentes, daß vom Wavemaster erstellt wird, befindet sich eine einfache Benutzeroberfläche, mit denen ein Anwender den Hyperwave-Server warten kann:

- **Login:** Der Benutzer kann sich identifizieren, um Daten in den Server einzuspielen, oder zu modifizieren
- **Edit:** Im Edit-Modus werden zusätzliche Möglichkeiten angeboten, um Dokumente am Server zu modifizieren.
- **Search:** Mit diesem Menüpunkt kann nach bestimmten Dokumenten am Server gesucht werden. Als Suchmöglichkeiten werden angeboten: Suche nach Titel, Schlüsselwörtern, Name, Erstellungsdatum, Autor oder Volltextsuche.
- **Annotate:** Mit einem Hyperwave-Server ist jeder identifizierte Benutzer in der Lage zu beliebigen Dokumenten seine Anmerkungen zu speichern.
- **Preferences:** Hier kann die bevorzugte Sprache und die gewünschte Qualität von Audio- oder Videodaten definiert werden.
- **Admin:** Für Benutzer mit Systemrechten besteht die Möglichkeit hier den gesamten Server zu administrieren.
- **Help:** Unter diesem Punkt befindet sich die Online-Hilfe von Hyperwave.

3.4 Hyperwave Publishing Wizard

Dies ist ein Programm, das es ermöglicht, sehr einfach eine größere Menge von Dokumenten vom File-System in den Hyperwave-Server zu laden.

Weiters ist es auch möglich, ganze Directory-Bäume, z.B. herkömmliche Web-Seiten sehr schnell in den Hyperwave-Server zu übernehmen.

3.5 Hyper-G Client Amadeus

Amadeus ist der erste Hyperwave PC-Client für das Betriebssystem Microsoft Windows. Er wurde entwickelt, um die Features von Hyperwave einfach nutzen zu können. Amadeus ist nicht nur ein **Viewer**, sondern ein vollständiges **authoring and document management tool** [Dietinger95].

3.5.1 Collection Browser

Der Collection Browser ist das entscheidende Tool von Amadeus. Mit seiner Hilfe ist es möglich die hierarchische Struktur der Daten am Hyperwave Server sehr einfach zu visualisieren. Zusätzlich ist eine Drag-and-Drop Funktionalität eingebaut, die das Kopieren und Verschieben von Daten am Server sehr vereinfacht.

3.5.2 Viewer

In Amadeus ist eine Reihe von internen und externen Viewern eingebaut, die eine Visualisierung von verschiedensten Dokumenttypen erlauben:

- TextViewer: PlainText, HTF (HyperText Format), HTML
- ImageViewer: GIF, JPEG, TIFF, PCX, BMP, XBM
- PostscriptViewer: Postscript 1 und Postscript 2
- SceneViewer (VRML-Viewer): SDF (Scene Description Format) und VRML
- MediaPlayer: MPGEG Video und Audio, AVI, MID, RMI, WAV, AU,...

4 MENSCH-MASCHINE-KOMMUNIKATION

4.1 Einleitung

4.1.1 Was ist Mensch-Maschine-Kommunikation (Human Computer Interaction)?

„Mensch-Maschine-Kommunikation beschäftigt sich mit dem Design von Computersystemen um den Bedürfnissen der Menschen zu entsprechen die mit diesen Systemen arbeiten.“ [Preece94].

Computer werden von sehr vielen und sehr unterschiedlichen Menschen benutzt. Daher ist es sehr wichtig Computersysteme so zu entwerfen, daß diese nicht nur von einigen wenigen Experten benutzt werden können, sondern von allen Personen die jemals auf diesem System arbeiten werden. Mensch-Maschine-Kommunikation beschäftigt sich mit dem Verstehen, Design, Evaluierung und Implementierung von interaktiven Computersystemen.

4.1.2 Warum Mensch-Maschine-Kommunikation?

Als 1950 die ersten kommerziell zu erwerbenden Computer auf dem Markt erschienen, waren sie nur von speziell eingeschulten Experten nutzbar. Die Gründe dafür lagen auf der Hand: Diese Geräte waren sehr teuer und im Vergleich dazu die Kosten für die Experten sehr gering. Sie wurden daher nur von einer kleinen Gruppe von Menschen (Techniker, Ingenieure, Wissenschaftler,...) genutzt, die auch in der Lage waren mit Lochkarten zu programmieren. Außerdem wußte noch niemand wie man die Bedienung vereinfachen könnte, bzw. erfolgten darüber noch keine Forschungen. Heute hat keiner dieser Gründe mehr Gültigkeit: Computer sind billig, sehr viele Menschen arbeiten damit und es gibt etliche Forschungsergebnisse wie man die Nutzung vereinfacht.

Durch die Entwicklung des Siliziumchips und die damit verbundene Miniaturisierung und Kostensenkung bei gleichzeitiger Rechengeschwindigkeit- und Speicherkapazitätssteigerung haben sich die Anwendungsmöglichkeiten vervielfacht.

Ein weiterer Meilenstein war die Entwicklung der Personal Computer (PC) in den 70er Jahren. Erst dadurch konnten Computer fast überall Einzug halten. Damit änderte sich aber auch die Benutzergruppe von wenigen, gut ausgebildeten Experten auf sehr viele Personen mit wenig Erfahrung im Umgang mit Computern. Genau hier setzt der Forschungsbereich Mensch-Maschine-Kommunikation an: Es wird daran gearbeitet,

Systeme zu entwickeln, die von jedermann ohne besondere Vorkenntnisse genutzt werden können. Weiters versucht man die Furcht oder Frustration, die bei der Arbeit mit diesem Medium entstehen kann, zu verringern. [Preece94] [Shneider92]

4.1.3 Mensch-Maschine-Kommunikation als fächerübergreifendes Forschungsgebiet

Der Bereich Mensch-Maschine-Kommunikation umfaßt einen großen Bereich an verschiedenen Forschungsdisziplinen [Preece94]:

- Computerwissenschaften
- Kognitive Psychologie
- Soziale und organisatorische Psychologie
- Ergonomie und menschliche Faktoren
- Ingenieurwissenschaften
- Design
- Anthropologie
- Soziologie
- Philosophie
- Linguistik
- Künstliche Intelligenz

Mensch-Maschine-Kommunikation läßt sich nach [Shneider92] in einzelne Forschungsrichtungen aufteilen:

- *Reduktion der Angst vor der Verwendung:* Computer sind schon sehr weit verbreitet, trotzdem werden sie nur von einem Teil der Bevölkerung genutzt. Viele Personen meiden aus Angst den Umgang mit ihnen. Vor allem ältere Menschen fürchten computerähnliche Geräte wie z.B. Bankomaten oder elektronische Schreibmaschinen weil sie Angst davor haben das Gerät zu beschädigen, einen schweren Fehler zu begehen, oder weil sie fürchten keinen Erfolg zu haben. Befragungen und Tests mit dieser Personengruppe zeigen Probleme auf die bei der Benutzung entstehen und lassen Lösungen dafür erkennen.
- *Sanfte Evolution:* Obwohl Anfänger ihre ersten Computererfahrungen bei der Arbeit mit Auswahlmenüs erleben, möchten sie später, wenn sie schon sicherer sind, schnellere und effizientere Möglichkeiten der Interaktion benutzen. Daher werden Methoden gesucht, die einen leichteren Übergang vom Anfänger zu einem erfahrenen Benutzer ermöglichen. Weiters muß man auf die unterschiedlichen Anforderungen von Anfängern und Experten achten. Eine Lösungsmöglichkeit wäre eine Reduktion der möglichen Interaktionen für Anfänger (nur die wichtigsten Interaktionen) und Experten, denen alle Möglichkeiten offen stehen sollen.
- *Menuauswahl und Eingabeformulare:* Der Inhalt, die Anzahl, die Platzierung und die Benennung von Menüpunkten haben einen großen Einfluß auf die Benutzbarkeit

eines Programmes. Schon mit relativ geringem Aufwand kann man große Erleichterungen erzielen. Die Einfachheit und richtige Anwendung von Eingabefeldern erlaubt schon Anfängern schnelle Erfolge bei der Erfassung von Daten mit einem Computer.

- *Kommandosprachen:* Sind ein traditioneller Interaktionsstil und damit ausgezeichnete Kandidaten für die Erforschung der Wichtigkeit von Konsistenz. Für Anfänger aber wenig geeignet, da sie eine gewisse Einlernphase erfordern, danach sind sie aber wesentlich effizienter als andere Interaktionsstile.
- *Direkte Manipulation:* Objekte werden visuell dargestellt und der Benutzer kann direkt mit ihnen arbeiten. Die Forschung versucht einerseits die Darstellung der Objekte zu verbessern und andererseits neue Formen der direkten Manipulation zu entwickeln (räumliche Visualisierung, Telepräsenz, virtuelle Realität,...).
- *Ein- und Ausgabegeräte:* In vielen Experimenten wird versucht die herkömmlichen Geräte zu verbessern und auch neue Formen der Eingabe zu finden (Datenhandschuh, räumlicher Joystick,...).
- *Erforschung von Information:* Dieser Bereich hat durch das Internet sehr stark an Bedeutung gewonnen. Man versucht die Probleme, die bei der Navigation in sehr großen Datenmengen entstehen, zu erkennen und zu beheben. Es werden neue Werkzeuge zur Datenvisualisierung entwickelt.
- *Online Hilfe:* Viele Computersysteme sind sehr komplex und bieten daher Unterstützung für den Benutzer in Form einer Hilfefunktion an. Das Forschungsziel ist eine so effiziente Unterstützung des Anwenders, daß er weder eine Einschulung noch ein Handbuch braucht, sondern selbstständig den Umgang mit dem System erlernen kann.
- *Spezifikation und Implementation von Interaktionen:* Die Ausarbeitung von Prinzipien, Regeln und Programmen um möglichst schnell und einfach gute Interaktionsmöglichkeiten für eigene Anwendungen zu gestalten.

4.1.4 Was ist ein Benutzerinterface?

Ein Benutzerinterface ist jener Teil eines Computersystems den der Benutzer sieht und mit dem er arbeitet. Das bedeutet, daß ein Benutzerinterface nicht nur ein visuelles Display ist, sondern die Summe aller Kommunikation zwischen Mensch und Computer. Es präsentiert Informationen dem Benutzer und akzeptiert Informationen (z.B. Tastatureingaben) vom Benutzer [Apple87].

4.1.5 Ziele und Rollenverteilungen bei der Entwicklung von Benutzerinterfaces

Die Ziele sind das Entwickeln bzw. Verbessern der Sicherheit, Nützlichkeit, Effektivität, Effizienz und Verwendbarkeit von Systemen [Preece94].

Laut [Shneider92] sollten folgende Ziele immer erreicht werden:

- *Standardisierung:* Für verschiedene Anwendungen sollte immer das gleiche oder zumindest ein sehr ähnliches Interface verwendet werden. Als Beispiel ließen sich hier die Apple Computer anführen, die schon sehr früh (1987) einheitliches Aussehens bei verschiedensten Programmen benutzen.
- *Konsistenz:* Bedeutet, daß übliche Aktionen, Ausdrücke, Layout, Farben und Typographie für das gesamte System gleich sind. So findet man z.B. bei Microsoft Windows in der Menuzeile ganz rechts immer den Menüpunkt „Datei“ („File“) und dort immer die Möglichkeiten eine Datei zu laden, zu speichern oder zu drucken.
- *Portabilität:* Ist die Möglichkeit Programme oder Daten auf verschiedensten Computersystem zu verwenden, ohne sie vorher mit großem Aufwand konvertieren zu müssen. Beispiele für gute Portabilität sind Anwendungen in der Programmiersprache Java oder Videodaten im MPEG-Format.

Mit Hilfe der Evaluierung (Beurteilung) des Systems besteht die Möglichkeit zu erfahren wie gut Anwender damit arbeiten können, bzw. welche Probleme wo auftreten.

Ausschlaggebende Kriterien sind laut [Shneider92]:

- *Einlernzeit:* Wie lange braucht ein typischer Anwender um mit dem Programm vernünftig arbeiten zu können, oder um bestimmte, oft auftretende Fälle zu bearbeiten?
- *Geschwindigkeit der Durchführung:* Wie lange braucht man um oft gewünschte Aufgaben lösen zu können? Bei einem Textverarbeitungsprogramm wäre es die Zeitdauer bis ein Brief geschrieben und ausgedruckt wurde.
- *Fehlerrate:* Wieviele Fehler werden bei der Ausführung der Aufgabe gemacht, und wie lange dauert es bis sie erkannt und behoben werden? Da die Zeit, die für Fehlerkorrekturen aufgewendet wird, sehr groß ist, wäre es wünschenswert, daß nur sehr wenige auftreten.
- *Merkbarkeit über die Zeit:* Wenn sich ein Anwender bei einem System eingelernt hat, und dann erst wieder nach einer Woche oder einem Monat mit dem System konfrontiert wird, wieviel hat er vergessen und muß wieder neu gelernt werden und wieviel wurde noch vom letztenmal gemerkt?

- *Subjektive Zufriedenheit:* Wie zufrieden war der Benutzer während er mit dem Programm gearbeitet hat? Dies läßt sich am besten durch Befragungen feststellen. Systeme bei denen die subjektive Zufriedenheit gering ist werden von Anwendern nur ungern und daher selten benutzt.

Die Rollenverteilung bei der Entwicklung von Benutzerinterfaces:

- *Anwender:* Ist der endgültige Benutzer der an dem System arbeitet.
- *Systemdesigner:* Entwirft das Gesamtkonzept und die Spezifikation.
- *Interfacedesigner:* Bestimmt die Aufgaben und Problemstellungen mit denen der Anwender konfrontiert wird und entwickelt daraus die Funktionsweise des Interfaces.
- *Softwaredesigner für die Funktionalität des Systems:* Implementiert alle Funktionen bis auf das Benutzerinterface.
- *Softwaredesigner für die Funktionalität des Benutzerinterfaces:* Programmiert die Teile, die dem Benutzerinterface zugrundeliegen.

Von der Größe des zu entwickelnden Systemes ist es abhängig, ob diese Rollen auf ein und dieselbe Person vereint werden, oder ob jede Rolle von einer speziell dafür geschulten Gruppe übernommen wird [Bass91].

4.1.6 Phasen der Softwareentwicklung

Der Lebenszyklus der Softwareentwicklung und damit auch der Entwicklung des Benutzerinterfaces kann in 6 Bereiche gegliedert werden [Bass91]:

- *Bedarfsdefinition:* Festlegung der Problemstellung.
- *Spezifikation:* Funktionelles und internes Design des Systems.
- *Implementation:* Programmierung des internen Designs.
- *Test:* Erfüllt das System die Aufgaben korrekt, die von ihm gefordert werden?
- *Installation:* Das System wird am Computer des Endanwenders plziert.
- *Wartung:* Auftretende Fehlfunktionen werden korrigiert und zusätzliche Wünsche des Anwenders werden in das System integriert.

In der Praxis werden diese Punkte oftmals nacheinander durchlaufen um die Qualität des Produktes zu verbessern.

4.2 Meilensteine in der Mensch-Maschine-Kommunikation

Theoretische Überlegungen, daß Computer für mehr verwendet werden können, als nur mathematische Berechnungen durchzuführen, wurden schon in den 60er Jahren angestellt. Dynabook, eine Idee von Alan Kay und Kollegen bei Xerox's Palo Alto Research Center in Kalifornien in den frühen 70ern, sollte ein Computer in der Größe eines dickeren Buches werden. Er sollte mit einem hochauflösendem Farbbildschirm, einer Funkverbindung in weltweite Datennetze, und einem Telefon ausgestattet sein. Weiters sollte er die Funktionen einer Sekretärin, eines Briefkastens, einer Bibliothek und auch die der Unterhaltung und Entspannung übernehmen. Aufgrund der damaligen Technologie konnte Dynabook nur eine Vision bleiben.

Aber in den späten 70er Jahren entwickelte die gleiche Gruppe ein Computersystem mit dem Namen „Star“ welches großen Einfluß auf die spätere Entwicklung des Personal Computers nahm: Ein knapp schreibtischgroßer Computer für immer nur einen Benutzer, ein großer Bildschirm, eine Maus und ein Benutzerinterface mit Fenstern und Icons bei dem mit der Maus Objekte direkt manipuliert werden können. Star benutzte auch die Philosophie des WYSIWYG (What You See Is What You Get). Sie bedeutet, daß zum Beispiel in einem Textverarbeitungsprogramm der Text genauso angezeigt wird wie er nach dem Ausdrucken auf einem Blatt Papier aussehen würde. Auf diesen neuen Ideen baute Apple auf und entwickelte den Apple Lisa in den frühen 80er Jahren.

Etwas später entstand das verbesserte Modell, der Apple Macintosh. Er übernahm die Grundideen vom Star-Computer, war aber aufgrund der inzwischen fortgeschrittenen Technologie wesentlich kleiner, billiger und schneller und wurde dadurch sehr erfolgreich. Alle graphisch orientierten Betriebssysteme, die danach entstanden, basieren bis heute auf den Prinzipien des Star und des Apple Macintosh [Preece94] [Apple87].

4.3 Menschliche Informationsverarbeitung

Dieser Bereich spielt eine fundamentale Rolle in der Mensch-Maschine-Kommunikation. Er liefert die theoretische Basis, wie die Informationsverarbeitung beim Menschen erfolgt, wie Wissen erworben wird, und welche Möglichkeiten existieren um das Lernen einfacher und effizienter zu gestalten.

4.3.1 Informationsspeicherung beim Menschen

Nach heutigem Wissensstand wird von einem Modell mit 3 verschiedenen Komponenten ausgegangen [Preece94]:

- *Sensorisches Gedächtnis*: Informationen der Umwelt werden hier registriert und für einige zehntel Sekunden gespeichert. Diese Gedächtnisform kann als Eingabebuffer für die direkte Repräsentation von Sensorinformation betrachtet werden. Nur ein kleiner Bruchteil der hier gespeicherten Information gelangt zur Weiterverarbeitung in das Kurzzeitgedächtnis. Die restlichen Daten werden ausgefiltert und wieder vergessen, bzw. von neu eingelangten Reizen „überschrieben“.
- *Kurzzeitgedächtnis*: Die Information die das sensorische Gedächtnis an das Kurzzeitgedächtnis weiterreicht wird hier verarbeitet und eventuell im Langzeitgedächtnis gespeichert. Heute wird der Begriff des Kurzzeitgedächtnisses durch den des Arbeitsgedächtnisses ersetzt, da in diesem Bereich jene Daten gespeichert werden, mit denen aktiv gearbeitet wird, wie z.B. eine Eingabe verarbeiten, eine Auswahl treffen, weitere Informationen holen und speichern, planen und Ausgabe (Sprache, Bewegung,...) vorbereiten. Die Hauptcharakteristik dieser Gedächtnisform besteht darin, daß die speicherbare Information in der Menge und in der Zeit begrenzt ist. Üblicherweise liegt die Grenze bei 5-9 gleichzeitig zu merkenden Einzelinformationen.
- *Langzeitgedächtnis*: Diese Informationen werden vom Kurzzeitgedächtnis zur Verfügung gestellt, unter Umständen werden sie sogar für den Rest des Lebens verfügbar gehalten und vom Kurzzeitgedächtnis auch wieder abgerufen.

4.3.2 Kognitive Modelle in der Mensch-Maschine-Kommunikation

Kognitive Prozesse sind Abläufe im menschlichen Gehirn, bei denen wir lernen Dinge zu verstehen, bzw. ganz allgemein wie wir Wissen erwerben [Preece94].

Das Modell der menschlichen Informationsverarbeitung war die Basis für die ersten Versuche in der Verbesserung der Mensch-Maschine-Kommunikation. Für eine gegebene Aufgabe war man der Meinung der Informationsaufnahme prozeß läuft in genau definierten Schritten ab. Daraus wurde das Modell des menschlichen Prozessors entwickelt. Diese ersten kognitiven Modelle schufen eine theoretische Basis um Vorhersagen über die Lerngeschwindigkeit beim Menschen zu treffen. Sie tendierten zu einer zu starken Vereinfachung des menschlichen Verhaltens, und werden heute durch verbesserte Modelle, die paralleles Denken beinhalten, ersetzt [Preece94].

4.3.3 Wahrnehmung und Repräsentation von Objekten

Das visuelle System des Menschen ist in der Lage Objekte bei extrem hellem Licht, in sehr dunklen Räumen und sich schnell bewegend Gegenstände zu erkennen. Aber trotzdem reagiert es nicht auf eine abgefeuerte Gewehrkegel oder infrarotes Licht. Dies bedeutet, daß wir einerseits einen sehr großen Bereich an angebotener Information

wahrnehmen können, aber ein weitaus größerer Teil an möglicher Information bleibt ohne technische Hilfsmittel (Zeitlupe, Infrarotkamera,...) für uns verborgen.

Im Laufe der Zeit haben sich zwei Theorien entwickelt, die versuchen die menschliche Wahrnehmung zu beschreiben [Preece94]:

- *Konstruktive Theorie:* Sie geht davon aus, daß der Prozeß des Sehens ein aktiver ist. Unsere Sicht der Dinge wird aus der Umgebungsinformation und dem gespeicherten Vorwissen konstruiert.
- *Ökologische Theorie:* Die Wahrnehmung beinhaltet nur den Prozeß des Informationssammelns aus der Umgebung und benötigt keine weiteren Prozesse der Konstruktion oder der Verfeinerung von Information.

4.3.4 Aufmerksamkeit und Beschränkungen bei der Informationsverarbeitung

Durch unsere Sinnesorgane werden ständig Informationen an das Gehirn geliefert. Da diese Informationsmenge viel zu groß ist, um sie zu verarbeiten, hat das menschliche Gehirn Methoden entwickelt um diese Flut an Neuigkeiten einzudämmen:

Eine Möglichkeit ist die Aufmerksamkeit nur auf eine bestimmte Sache oder Person zu lenken. Dies ist der Fall, wenn man sich mit vielen Personen gleichzeitig in einem Raum befindet, sich mit einer angeregt unterhält und alles andere um einen herum kaum wahrnimmt.

Eine zweite Möglichkeit ist die geteilte Aufmerksamkeit: Man arbeitet konzentriert an einer Sache und beschäftigt sich nebenbei mit etwas ganz anderem (z.B. gleichzeitig Auto fahren und sich mit jemanden unterhalten). Obwohl beide Prozesse ein großes Maß an Informationsverarbeitung beinhalten, ist das Gehirn in der Lage beides gleichzeitig auszuführen [Preece94].

Im Bereich Benutzerinterface gibt es viele Beispiele, die diese beiden Methoden der Informationsfilterung benutzen:

Graphische Benutzerinterfaces, die meist optisch sehr gut strukturiert sind, lenken die Aufmerksamkeit des Benutzers auf das Interface. Dadurch kann er sich besser der Lösung seiner Aufgabe widmen, da Ablenkungen, wie z.B. die Suche in Handbüchern nach einem bestimmten Befehl sehr stark reduziert werden.

Neuere Betriebssysteme erlauben das *Multitasking*. Das bedeutet ein Benutzer kann mehrere Aufgaben gleichzeitig von einem Computer bearbeiten lassen. Dies zielt in den Bereich der geteilten Aufmerksamkeit. In der Praxis entstehen aber oft Probleme: Es wird an einer Aufgabe „A“ gearbeitet, nach einer gewissen Zeit wird zu der zweiten Aufgabe „B“ übergegangen und wenn dann wieder zur ersten Aufgabe „A“ gewechselt

wird, hat der Benutzer schon meistens vergessen woran er gearbeitet hat, wieviel er schon erledigt hat und wieviel er noch erledigen muß. Im täglichen Leben wird diesem Problem mit kognitiven Hilfen (Notizzettel, Post-it, der sogenannte Knoten im Taschentuch,...) begegnet. Bei der Arbeit am Bildschirm sind diese Möglichkeiten allerdings nicht sinnvoll und daher müssen neue Methoden entwickelt werden. Eine mögliche Lösung wäre eine Art Statusfenster für jede Aufgabe das genaue Informationen darüber enthält, welche Arbeiten schon ausgeführt wurden und welche Arbeiten noch zu erledigen sind, sozusagen ein virtueller Notizzettel der automatisch vom Computer erstellt wird.

Ein weitere Einschränkung des menschlichen Gehirns ist, das es Informationen wesentlich leichter wiedererkennen kann, als die gleichen Informationen aus dem Gedächtnis abzurufen. Auch dies kann mit graphischen Benutzeroberflächen berücksichtigt werden: Es fällt Menschen wesentlich leichter Icons (das sind kleine Graphiken, die eine ganz bestimmte Bedeutung haben, z.B. die Zeichen an der Türe zur Unterscheidung zwischen Herren- und Damentoilette) und ihre Bedeutung wiederzuerkennen, als sich ein bestimmtes Kommando aus dem Gedächtnis abzurufen. Als klassisches Beispiel kann der Unterschied angeführt werden wie in einem kommandoorientierten Betriebssystem (z.B. MS-DOS bei PCs) und einem graphischen (z.B. Apple Macintosh oder Microsoft Windows95) eine Datei gelöscht werden kann:

Beim kommandoorientierten Systemen muß sich der Benutzer einen Befehl merken, in diesem Fall „delete“ beim graphischen klickt er die Datei mit der Maus an und zieht sie zu einem virtuellen Papierkorb in dem diese Datei dann verschwindet. Wenn dieses Prinzip des „Wiedererkennen anstatt Erinnern“ konsequent angewandt wird, kann es die Benutzung eines Computers oder eines Programmes wesentlich vereinfachen.

4.3.5 Wissen und mentale Modelle

In der Erforschung der Mensch-Maschine-Kommunikation versucht man durch neue Erkenntnisse über das menschliche Wissen und mentaler Modelle die Einlernzeit, die oft auftretenden Fehler und die Einfachheit mit der Benutzer eine gestellte Aufgabe lösen, vorherzusagen. Weiters wird versucht, den Prozeß des Einarbeitens in ein neues System zu analysieren und zu vereinfachen. Ein Forschungsziel ist es zu erkennen wie Menschen ihr Wissen organisieren und wie mentale Modelle bei Benutzerinterfaces eingesetzt werden können [Preece94].

Die Repräsentation menschliches Wissen kann in drei Typen aufgeteilt werden:

- *analoge Repräsentationen:* sind Bilder ähnlich wie Photos von einem Gegenstand.
- *aussagende Repräsentationen:* abstrakte, sprachähnliche Konstrukte, z.B. „das Bild hängt an der Wand“ oder „die Türe ist geschlossen“.

- *verteilte Repräsentationen*: Netzwerke von Knoten, die Information befindet sich implizit in den Verbindungen zwischen den einzelnen Knoten.

Das Wissen des Menschen ist sehr stark organisiert. Dies läßt sich anhand eines einfachen Beispiels beschreiben: Im Gedächtnis ist eine sehr große Anzahl von Informationen gespeichert und trotzdem ist es sehr leicht möglich bestimmte Fakten innerhalb sehr kurzer Zeit abzurufen:

Es bereitet niemandem Probleme sofort die Hauptstadt von Frankreich zu nennen oder eine Frucht deren Name mit dem Buchstaben „A“ beginnt. Die schnellen Antwortzeiten für diese Fragen deuten darauf hin, daß das menschliche Wissen organisiert sein muß. Wie dieses Wissen allerdings genau organisiert wird kann erst in Ansätzen vermutet werden:

Einer davon ist das *semantische Netzwerk*: Es wird davon ausgegangen, daß Konzepte miteinander verbunden sind, z.B. die Namen von Staaten und deren Hauptstädten werden verteilt gespeichert, aber es besteht eine starke Verbindung zwischen jedem Staatsnamen und der dazugehörigen Hauptstadt, sodaß es relativ einfach ist zu einem Staat auch seine Hauptstadt zu nennen.

Eine weitere Theorie vermutet, daß das Wissen in Form von *Schematas* gespeichert wird: Ein Schema ist in diesem Fall ein Netzwerk von Allgemeinwissen das sich aus bisherigen Erfahrungen ergab. Schemas können in *Scripts* unterteilt werden die den genauen Ablauf eines ganz bestimmten Szenarios beschreiben, wie z.B. der Verlauf eines Essens in einem Restaurant. Zuerst wird das Lokal betreten, ein Platz zum Sitzen ausgesucht, die Speisekarte betrachtet, bestellt, gegessen, bezahlt und dann wird wieder das Lokal verlassen. Diese Scripts dienen dazu die täglichen Abläufe des Lebens einfach durchführen zu können. Es sind sozusagen erlernte Verhaltensweisen die immer gleich ablaufen. Interessant wird es wenn diese Scripts absichtlich durch äußere Einflüsse unterbrochen werden. Es werden fast alle Menschen zunächst einmal völlig verwirrt sein wenn der Kellner sofort nach der Aufnahme der Bestellung die Rechnung präsentiert und abkassieren will, und dann erst das Essen bringt. Daran lassen sich die Nachteile dieser erlernten Schematas erkennen: Passiert etwas völlig Unerwartetes, das eben nicht ins Schema paßt, muß erst das gerade laufende Script verworfen und dann improvisiert werden. Dieses Problem der Schematas und Scripts läßt sich auch auf den Computer anwenden. Zum Beispiel beim Apple Macintosh oder unter Windows95 von Microsoft verläuft das Speichern von Daten immer nach einem gleichen Schema. Ändert nun ein Programm den Ablauf des Abspeicherns, oder wechselt der Benutzer von einem System zu einem das sich völlig anderes verhält, wirkt sich das bisher erlernte Script fatal aus: Der Benutzer weiß zwar wie er es bisher gemacht hat, er weiß auch genau was er will, aber er muß erst herausfinden, wie er das an diesem neuen System bewerkstelligen kann. Dieses Problem ist einer der Hauptgründe warum

Menschen einem neuem System, das das alte System und die erlernten Scripts ersetzen soll sich so ablehnend verhalten. Eine Möglichkeit dieses Problem zu lösen bestünde darin, gewisse Operationen bei allen Betriebssystemen zu vereinheitlichen. Die Ansätze dazu sind glücklicherweise schon vorhanden.

Der größte Kritikpunkt an Schematas und Scripts liegt in deren Inflexibilität. Für die täglichen Situationen des Lebens sind sie recht praktisch, bei komplexeren oder bei neuen Situationen haben sie überhaupt keinen Nutzen. In diesen Fällen greift der Mensch auf mentale Modelle zurück. Der Vorteil eines solchen Modelles liegt darin, daß man damit gedanklich experimentieren kann, d.h. man kann sich überlegen was passiert wenn man etwas Bestimmtes ausführt, bzw. was passiert wenn man es nicht ausführt. Mentale Modelle sind eine Kombination von analogen und aussagenden Repräsentationen. Sie sind nicht nur ein Bild von einer Sache, sondern die Ansicht davon. Dadurch bieten sie den Vorteil, daß der Mensch in der Lage ist Vorhersagen über die zukünftige Entwicklung eben dieser Sache zu tätigen. Mentale Modelle haben allerdings auch einen großen Nachteil der sich leicht demonstrieren läßt: Man stelle sich ein Haus mit einer elektrischen Heizung vor. Diese Heizung wird durch ein Thermostat geregelt. Das Haus wurde eine Zeitlang nicht geheizt weil deren Bewohner auf Urlaub waren. Wenn sie wieder nachhause kommen, merken sie, das es nur 10 Grad Celsius hat, sich alle Bewohner aber erst bei 23 Grad Celsius wohl fühlen. Es wird meistens folgendes passieren: Das Thermostat der Heizung wird nicht auf 23 Grad eingestellt, sondern auf das maximal mögliche, in der Hoffnung daß es dadurch schneller wärmer wird. In Wirklichkeit ist es aber vollkommen egal ob es auf 23 Grad oder auf z.B. 35 Grad eingestellt wird, dadurch wird es nicht schneller wärmer. Der Grund liegt darin, daß die Heizung durch ein Thermostat nur ein- bzw. ausgeschaltet werden kann. Deswegen ist es völlig sinnlos, das Thermostat auf das Maximum einzustellen.

Daß sich Menschen aber trotzdem so verhalten liegt in diesem Beispiel daran, daß die elektrische Heizung noch mit einer älteren Form (Holz, Kohle, Öl) verglichen wird. Bei diesen alten Formen lautete das Prinzip „je mehr Brennmaterial, desto schneller heißer“. Daran kann man erkennen, daß mentale Modelle unter gewissen Umständen nicht zur Lösung eines Problem es beitragen können, bzw. können sie das Verständnis für eine richtige Lösung des Problem es noch viel mehr erschweren, da Menschen eher bereit sind ein Schema oder Script umzulernen als ein mentales Modell. Um auf das vorherige Beispiel mit dem Haus und der Heizung zurückzukommen: Bisher wurde noch keine befriedigende Lösung gefunden, um das Aufdrehen des Thermostates auf das Maximum zu verhindern. Es ist sogar nicht einmal möglich den Menschen zu erklären, daß dieses Verhalten absolut keinen Sinn ergibt.

Mentale Modelle lassen sich in zwei Untergruppen auftrennen:

- *strukturelles Modell* („wie funktioniert das“): Beschreibt die inneren Mechanismen eines Gerätes anhand dessen Komponenten. Mit diesem Modell ist ein Benutzer in der Lage Vorhersagen über das Verhalten eines Gerätes zu tätigen. Bei der Reparatur eines nicht mehr funktionierenden wird immer dieses mentale Modell genutzt. Es ist sehr schwierig nur mit Hilfe des strukturellen Modelles die Benutzung eines Gerätes abzuleiten. Andererseits braucht man zur Bedienung eines Gerätes nicht unbedingt diese Informationen, dafür reicht das nachfolgende Modell aus:
- *funktionelles Modell* („wie benutzt man das“): Dieses liefert nur Informationen über die Nutzung eines Gerätes und nichts über den inneren Aufbau oder die Funktionsweise. Viele Menschen können etwas Neues allein mit Hilfe dieses Modelles benutzen ohne eine Ahnung von der inneren Funktionsweise zu haben.

Diese beiden Modelle lassen sich leicht unterscheiden: Jeder der ein Telefon hat, weiß genau wie er es benutzen muß um eine andere Person anzurufen. Dieses Wissen ist das funktionelle Modell. Aber nur sehr wenige wissen, wie ein Telefon genau funktioniert. Diese Personen haben neben dem funktionellem Modell auch ein Wissen über das strukturelle Modell. Deswegen sind auch nur sie in der Lage, Fehler zu erkennen, zu analysieren und zu beheben.

Zusammenfassend läßt sich sagen, daß zur Benutzung eines Systems das funktionelle Modell vollkommen ausreicht, um aber Fehler zu beheben oder das System zu verbessern, ist unbedingt das strukturelle Modell notwendig. Aus diesem Grund besteht das Wissen, das Menschen über ein System haben, fast immer nur aus einem funktionellem Modell.

4.3.6 Interfacemetapher und Begriffsmodelle

Der Begriff Metapher [Preece94] wird aus traditionellen Gründen mit der Sprache assoziiert. Immer wenn Menschen ein abstraktes Konzept zu erklären versuchen, verwenden sie bekannte, für jeden verständliche Begriffe, sogenannte metaphorische Ausdrücke. Als ein sehr gutes Beispiel für die Verwendung von Metaphern kann ein Gespräch über die Zeit dienen: Zeit ist ein sehr abstrakter Begriff und so haben die Menschen als Metapher dafür das Geld gewählt: man kann „Zeit sparen“, „Zeit verbrauchen“, „Zeit verschwenden“, „jemanden seine Zeit geben“, „mit geborgter Zeit leben“.

Die gesamte menschliche Sprache basiert auf den verschiedensten metaphorischen Abstraktionen. Deswegen werden auch in der Mensch-Maschine-Kommunikation gerne Metapher für abstrakte Begriffe verwendet: Die Benutzeroberfläche graphischer

Systeme wird oft als „Desktop“ (Schreibtisch) bezeichnet, es gibt Menüs, Fenster (windows), man kann Texte ausschneiden (cut), kopieren (copy) und einfügen (paste).

Verbale Metapher werden immer dann verwendet, wenn Menschen eine neue Technologie erlernen, wie z.B. einen Computer: Zunächst versuchen sie eine Metapher zu finden mit der sich ein Computer vergleichen läßt. Meistens endet dieser Vergleich in der Metapher „Schreibmaschine“, bzw. „Computer sind so ähnlich wie Schreibmaschinen“. Mit Hilfe dieser Erkenntnis können Menschen dann etwas über das Verhalten des Computers aussagen, wenn sie zum Beispiel eine Taste auf der Tastatur drücken. Sie sind mit dieser Metapher sogar in der Lage einen einfachen Brief in einer Textverarbeitung zu schreiben. Das bedeutet also, die Metapher „Schreibmaschine“ hat ihnen geholfen das Gerät „Computer“ ein wenig zu verstehen und damit auch bedienen zu können. Nachdem diese Menschen eine Zeitlang mit dem Computer gearbeitet haben, werden sie aber feststellen, daß die Metapher „Schreibmaschine“ nicht ausreichend ist, um alle Möglichkeiten eines Computers zu nutzen:

Es ist mit einer Schreibmaschine nicht möglich, einen Text zu speichern, Teile des Textes zu löschen, zu kopieren, einzufügen, den Text anders zu formatieren bzw. die Schriftgröße zu ändern. Gleichzeitig mit dieser Erkenntnis beginnen die Menschen die Metapher „Schreibmaschine“ für den Computer zu verwerfen, und versuchen eine neue Metapher „Computer“ aus ihren bisherigen Erkenntnissen zu entwickeln. Nachdem diese Personen sich auch mit Tabellenkalkulationsprogrammen, Spielprogrammen, usw. beschäftigt haben, wird die Metapher „Schreibmaschine“ für Computer völlig verworfen und stattdessen die Metapher „Computer“ ständig weiterentwickelt und verbessert. Mit dieser erlernten „Computermetapher“ sind diese Menschen dann auch in der Lage, sich auf anderen oder neuen Computersystemen relativ schnell einzuarbeiten, weil sie die Lernphase der Metapher „Schreibmaschine“ schon durch die bessere des „Computers“ ersetzt haben.

Zusammenfassend läßt sich erkennen, daß bei einer neuen Technologie zuerst eine ähnliche, schon bekannte Technologie zur Erklärung verwendet wird, um das Neue zu erforschen, dann aber wird aus den gewonnenen Erkenntnissen ein neues gedankliches Modell gebildet.

Von Xerox wurde erstmals der Begriff der *virtuellen Interfacemetapher* verwendet. Ihr System (Star) versuchte den Computer als einen Schreibtisch wie in einem Büro darzustellen. Man hatte eine Tischfläche, die durch den Bildschirm repräsentiert wurde. Auf dieser Tischfläche befanden sich Icons für die verschiedenen Dokumente und Ordner. Durch einfaches Anklicken eines Dokumentes mit der Maus konnte das Dokument geöffnet und bearbeitet werden. Durch Verschieben der Dokumente in Ordner wurden sie darin abgelegt. Es existierte sogar eine Metapher für einen Papierkorb: Ein Icon das die Form eines Papierkorbes hatte. Dokumente und Ordner die

man nicht mehr brauchte, wurden mit der Maus zum Papierkorb-Icon gezogen und damit auch physikalisch vom Computer gelöscht. Dieser erste Versuch die Metapher „Büroschreibtisch“ für Computer einzuführen hatte in den frühen 80er Jahren einen so großen Erfolg, daß diese Metapher sogar noch heute verwendet wird (Apple Macintosh, Microsoft Windows, X-Windows unter Unix). Allerdings stellte man schon damals fest, daß auch diese „Schreibtischmetapher“ nur unzureichend ist, denn bisher gibt es noch keinen Tisch, bei dem die daraufliegenden Akten durch einen Knopfdruck zusammengeräumt und vielleicht auch noch alphabetisch sortiert auf einer Seite des Tisches gestapelt werden. Auch gibt es noch keinen Ordner der ein Dokument, welches man darauflegt, automatisch einheftet. An diesen Beispielen erkennt man, daß auch diese Metapher „Büroschreibtisch“ Unzulänglichkeiten hat. Trotzdem wurde sie von vielen Menschen akzeptiert und hat auch die Einlernzeit und die Fehler bei der Nutzung von Computern stark reduziert.

Vergleich von physikalischen Aktionen mit Mausektionen bei der „Schreibtischmetapher“:

Mausaktion	physikalische Aktion
zeigen (point)	mit dem Finger auf etwas zeigen
anklicken (click)	mit der Hand das gewünschte Objekt berühren
auswählen (select)	das Objekt in die Hand nehmen
verschieben (drag and drop)	in die Hand nehmen, halten, woandershin tragen und fallenlassen

4.3.7 Lernen in Zusammenhängen

Wenn Menschen zum ersten Mal vor einem Computer sitzen, ist ihre Reaktion meistens Angst und Verunsicherung [Preece94]. Sitzen aber die gleichen Menschen zum ersten Mal hinter dem Lenkrad eines Autos sind sie hochmotiviert und wollen gleich losfahren. Warum gibt es in diesen beiden Situationen so große Gefühlsunterschiede? Ein Grund liegt sicherlich darin, daß beim Autofahren der Schüler sofort zu starten probiert, dann anfährt, schaltet, und weiterfährt. Vom Fahrlehrer werden nur wenige Erklärungen gebracht und der Schüler muß sich auch vorher keine Bedienungsanleitung durchlesen. Das Lernen erfolgt dadurch das der Schüler einfach losfährt. Erste Erfolge sind sofort da, und nach wenigen Stunden kann der Schüler schon einigermaßen gut fahren. Das bedeutet, daß ein sehr komplexer Prozeß wie das Autofahren in einigen Stunden gelernt werden kann. Im Laufe der Zeit muß der Schüler noch seine Fähigkeiten verbessern und

Erfahrungen sammeln, aber der grundsätzliche Umgang mit dem Auto wurde sehr schnell gelernt. Im krassen Gegensatz dazu verhält sich das Lernen des Umgangs mit Computern: Der Schüler muß sich durch schlechtgeschriebene Handbücher arbeiten, ärgert sich mit ungenau beschriebenen Beispielen herum, und ist weit davon entfernt, nach wenigen Stunden einen Computer einigermaßen bedienen zu können. Einlernzeiten bis zu einem halben Jahr um nur ein bißchen damit arbeiten zu können sind sehr wahrscheinlich. Außerdem ist in dieser Zeit die Anzahl der Mißerfolge wesentlich größer als die der Erfolge. Die Folgen die daraus entstehen sind meist eine subjektive Unzufriedenheit oder ein unangenehmes Gefühl bei der Arbeit mit dem Computer.

Lernschwierigkeiten bei Computern:

- *Lernen ist schwierig:* Lernende erfahren Frustration und blamieren sich; das Lernen dauert länger als erwartet, und Lernende haben Probleme ihr neuerworbenes Wissen auch anzuwenden.
- *Lernenden fehlt Basiswissen:* Lernende wissen nicht wie man mit Computer umgeht, sie verstehen auch nicht den „Computer-Jargon“ der in vielen Handbüchern verwendet wird; Lernende wissen nicht, was relevant ist um Probleme zu verstehen und zu lösen.
- *Lernende machen „ad hoc“ Interpretationen:* sie versuchen zu interpretieren was sie gerade tun, bzw. was ihnen gerade passiert; Interpretationen verdecken die Sicht auf das eigentliche Problem und den richtigen Lösungsweg.
- *Lernende generalisieren aus ihrem bisherigen Wissen:* z.B. glauben sie, eine Textverarbeitung arbeitet gleich wie eine Schreibmaschine, daraus folgen Mißverständnisse und Probleme.
- *Lernende haben Schwierigkeiten Anweisungen zu folgen:* Lernende lesen nicht immer Anweisungen oder führen sie nicht immer aus; sie verstehen Anweisungen oft nicht oder führen sie nicht korrekt aus.
- *Abhängigkeit von Problemen:* Sie haben Schwierigkeiten zu erkennen, daß ein Problem weitere Probleme auslösen kann.
- *die Möglichkeiten die das Interface anbietet sind nicht offensichtlich:* Lernende werden durch Seiteneffekte und Rückmeldungen des Computers verwirrt.
- *die angebotene Hilfe des Computers hilft nicht wirklich:* Lernende wissen nicht immer welche Hilfe sie brauchen würden; die angebotene Hilfe ist nicht auf das aktuelle Problem des Benutzers zugeschnitten.

Wenn Menschen mit einem neuen System konfrontiert werden, verwenden sie ihr bisheriges Wissen um das Verhalten des neuen Systems zu interpretieren und vorherzusagen. Lernen durch Analogie versucht eine Brücke zu schlagen zwischen dem

bisherigen Wissen und dem Neuen. Zum Beispiel muß sehr oft eine Schreibmaschine als Analogie für eine Textverarbeitung dienen. Interfacemetapher und „mainstream instructions“ wie z.B. Laden, Speichern, Drucken einer Datei bauen ebenfalls auf dem Prinzip Lernen durch Analogie auf.

Immer wenn Menschen etwas Neues lernen, machen sie auch Fehler. Meistens ist das nicht problematisch, es liefert nur ein Feedback, daß etwas falsch gemacht wurde. Aber bei Computersystemen fürchten Menschen sich davor Fehler zu machen. Einerseits weil sie glauben zu dumm für den Computer zu sein, und andererseits haben sie Angst etwas zu zerstören (Daten oder das Gerät selbst). Aufgrund dieser Befürchtungen sind Menschen stark gehemmt mit einem Computer herumzuprobieren und zu schauen was geschieht wenn sie einen Befehl ausführen. Diese Hemmschwelle bewirkt aber, das sie nur sehr schwer den Umgang mit diesen Geräten erlernen.

Fehler, die Benutzer auslösen, werden grob in zwei Punkte unterteilt:

- *Entscheidungsfehler*: entstehen durch eine falsche Entscheidung in der Nutzung des Programmes: z.B. ein Benutzer verschiebt das Icon für eine Festplatte auf den Papierkorb und glaubt, damit den Inhalt der Festplatte zu löschen. In Wirklichkeit muß er für diese Operation aber einen bestimmten Menüpunkt auswählen. Damit der Benutzer diese Art von Fehlern beheben kann, muß er meist zusätzliche Informationen aus einer Online-Hilfe, einem Handbuch oder einem Kollegen einholen. Die Behebung dieser Fehler nimmt sehr viel Zeit ein.
- *Unachtsamkeitsfehler*: entstehen dadurch, daß der Benutzer nicht genügend konzentriert bei der Arbeit ist: Als Beispiele lassen sich dafür Tippfehler, oder ein falsch angewählter Menüpunkt aufzählen. Allen diesen Fehler ist gemeinsam, daß der Benutzer genau weiß wie er es richtig ausführen muß und daher meistens sehr schnell den Fehler beheben kann.

In qualitativ hochwertigen Benutzerinterfaces wird versucht, die Anzahl der Fehler die ein Benutzer machen kann, möglichst gering zu halten, bzw. die Fehlermeldungen klar und verständlich zu formulieren. Es ist allerdings darauf zu achten, daß Menschen gewohnt sind aus Fehlern zu lernen. Scharf formuliert würde es dann bedeuten: Wenn jemand in einem System keine Fehler machen kann, kann er auch nichts über das System lernen.

4.4 Gestaltung von Benutzerinterfaces

4.4.1 Allgemeines

Nach [Preece94] gelten folgende Punkte für das Design von Benutzerinterfaces:

- Design ist ein kreativer Prozeß, aber trotzdem müssen Entwicklungsprinzipien beachtet werden.
- Design involviert das Erzeugen einer Anzahl von zwischenzeitlichen Repräsentationen des endgültigen Produktes.
- Benutzerbezogenes Systemdesign zentriert sich auf Menschen, ihre Arbeit, ihre Umgebung und wie das neue System am besten designed werden kann um diese Menschen zu unterstützen.
- Benutzerbezogenes Design benötigt benutzerbezogene Repräsentationen von Objekten.

Designprinzipien (principles) und Designregeln (guidelines) [Preece94] können folgendermaßen unterschieden werden:

- Prinzipien sind abstrakt und auf verschiedenste Gebiete anwendbar, Designregeln entsprechen einfachen Instruktionen.
- Prinzipien müssen interpretiert und in Relation zum aktuellen Problem gestellt werden.
- Designregeln basieren auf psychologischen Erkenntnissen und auf praktischer Erfahrung.

Zusätzlich zu den Prinzipien und Regeln für Interfacedesign wurden von [Apple87] noch Strategien erstellt, deren Gültigkeit unabhängig von Computersystemen und Programmen ist:

- *keine unterschiedlichen Moden (modelessness)*: „Eine vom Benutzer ausgeführte Aktion sollte immer das gleiche Ergebnis liefern, unabhängig von den vergangenen Aktivitäten. Wenn trotzdem verschiedene Moden verwendet werden, müssen diese deutlich gekennzeichnet werden.“ Ein Modus ist ein Kontext, in dem die Aktion eines Benutzers verschieden von einem anderen Kontext interpretiert wird. Im wirklichen Leben verwenden Menschen keine Moden, daher geben verschiedene Moden bei Computern ihnen das Gefühl, daß Computer „unnatürlich“ und „unfreundlich“ sind. Als Beispiel für einen Mode, der immer wieder Probleme bei Anfängern verursacht, gilt die Verwendung des Scrollbars bei einem Fenster: Um die Scrollbar zu bewegen muß der Benutzer die Maustaste gedrückt halten während er die Maus bewegt. Aus seinen bisherigen Erfahrungen weiß er aber nur, daß die Maus zum Zeigen und Anklicken verwendet werden kann. Obwohl verschiedene Moden so problematisch sind, gibt es doch auch eine Anwendung die dieses Problem sehr elegant löst und wo auch fast nie Schwierigkeiten auftreten: Bei Zeichenprogrammen kann man mit der Maus z.B. einen Pinsel oder eine Spraydose auswählen und mit dieser dann am Bildschirm zeichnen. Es besteht aber auch die

Möglichkeit einen Radiergummi zu selektieren um bestimmte Bereiche des Bildes zu löschen. Das bedeutet, daß der Mauszeiger die unterschiedlichsten Moden (zeichnen, sprayen, löschen,...) annehmen kann, und trotzdem sind die Benutzer überhaupt nicht verwirrt. Da der Mauszeiger in jedem Modus eine genau definierte Form annimmt, Pinsel zum Malen, Spraydose zum sprayen, Radiergummi zum löschen, stellen die Menschen eine Analogie zur wirklichen Welt her: Der Mauszeiger ist ihre verlängerte Hand, und diese hält z.B. einen Pinsel.

- *Ereignisschleife (event loop)*: „Anwendungen erlauben dem Benutzer alles zu jedem beliebigen Zeitpunkt durchzuführen.“ Jedes Programm befindet sich in einer Warteschleife und der Benutzer ist jederzeit in der Lage jeden beliebigen Befehl auszuführen. Obwohl diese absolute Freiheit sehr angenehm ist, hat sie doch auch einen Nachteil: Ein Anfänger weiß nicht in welcher Reihenfolge er die Menüpunkte aufrufen soll um sein Problem zu lösen. Deswegen wurden Assistenten (agents, wizards) entwickelt, die es dem Anfänger erlauben, auch komplexe Probleme leicht zu lösen. Die Aufgabe des Assistenten ist es ihn nach seinen Wünschen zu fragen und dann daraus eine entsprechende Lösung zu ermitteln. Ein Beispiel wäre in Microsoft Word for Windows der Lebenslauf-Agent, der gezielte Fragen stellt und dann aus den Antworten einen optisch ansprechenden Lebenslauf erstellt.
- *zurücknehmen von Aktionen (reversible actions)*: „Immer einen Weg zurück zur Verfügung stellen.“ Da durch die Ereignisschleife der Benutzer aufgefordert wird zu experimentieren, er sich aber vor der Vernichtung seiner bisherigen Arbeit fürchtet, muß ihm die Möglichkeit geboten werden, alle Änderungen an seinen Daten rückgängig zu machen. Wird diese Möglichkeit nicht angeboten, haben Benutzer Angst davor eine Funktion eines Programmes auszuprobieren.
- *Bildschirm (screen)*: „Der Bildschirm ist die Bühne der Mensch-Maschine-Kommunikation.“ Am Bildschirm sollte für den Benutzer immer leicht erkennbar sein, was der Computer gerade ausführt. Bei längeren Operationen sollte immer eine Anzeige sichtbar sein, die den aktuellen Status, und die ungefähre Zeit bis zum Ende der Operation anzeigt. Es ist sehr wichtig, daß nicht nur der prozentuelle Verlauf sondern auch die noch abzuwartende Zeit angezeigt wird, da nur aus einem prozentuellen Verlauf keine Aussage über die zu verbleibende Wartezeit getätigt werden kann.
- *einfache Sprache (plain language)*: „Die Kommunikation mit dem Benutzer erfolgt in klaren und einfachen Sätzen.“ Es ist sehr wichtig, daß Meldungen, besonders Fehlermeldungen, klar und einfach formuliert werden und keine technischen Fachausdrücke beinhalten.
- *testen des Benutzerinterfaces*: „Nur der Test eines neuen Produktes kann seine Qualität bestätigen.“ Im Verlauf des Designprozesses sollten immer wieder

Außenstehende in Tests involviert werden, um zu erkennen ob das Interface den gewünschten Anforderungen entspricht.

- *Beachtung von Behinderungen an Menschen:* Für Personen die schlecht sehen sollte eine Möglichkeit vorgesehen werden, mit denen sie die Schriften und Graphiken vergrößern können. Information darf nicht nur farbkodiert angeboten werden, da einige Personen farbenblind sein könnten, und dann diese Daten nicht interpretieren können. Keine Meldung des Systems sollte nur durch Audiosignale erfolgen. Mindestens muß die Möglichkeit angeboten werden, eine zusätzliche visuelle Anzeige einzuschalten.

4.4.2 Unterschiedliches Design für unterschiedliche Anwendungen

Bei Computersystemen kann man zwischen 4 Gruppen unterscheiden [Shneider92]. Die Unterschiede liegen in der Einlernzeit, der Komplexität des Systemes, der möglichen Fehler und der subjektiven Zufriedenheit des Benutzers:

- *lebenskritische Systeme:* Sind Luftüberwachungssysteme für Flugzeuge, Kontrollsysteme für Kernreaktoren, Energieversorgungssysteme, bemannte Raumfahrt, militärische Operationen und medizinische Anwendungen. All diesen Systemen ist gemeinsam, daß sie sehr teuer, zuverlässig und effektiv sind. Die Benutzer brauchen eine lange Einlernzeit um auch unter Streß fehlerfrei zu arbeiten. Die subjektive Zufriedenheit ist egal, da die Personen hochmotiviert sind. Mit ihnen werden auch immer wieder als Übung die verschiedensten Extremsituationen durchgespielt.
- *industrielle und kommerzielle Anwendungen:* Typische Anwender sind Banken, Versicherungen, Fluglinien, Hotels und Kreditkartenfirmen. Sie legen Wert auf geringe Kosten, schnelle Verarbeitungszeiten, weniger Wert auf Einlernzeit und überhaupt keinen auf die subjektive Zufriedenheit ihrer Mitarbeiter.
- *Büro-, Heim- und Unterhaltungsanwendungen:* Beispiele dafür sind Textverarbeitungen, Tabellenkalkulationen, Unterhaltungsprogramme und Spiele. Wichtig ist hier eine möglichst kurze Einlernzeit, geringe Fehlerrate, kleiner Preis und subjektive Zufriedenheit der Benutzer. Die Schwierigkeit im Design dieser Systeme liegt darin, daß die Anwender unterschiedlichstes Vorwissen über Computer haben und trotzdem sollen ein absoluter Anfänger und ein Experte mit dem System zufrieden sein.
- *Forschungs-, Kreativitäts-, und Kooperationsanwendungen:* Das sind Datenbanken, Visualisierungen großer Datenmengen, Expertensysteme, Diagnosesysteme, Designsysteme, kooperative und kommunikative Systeme. In diesem Bereich sind die Benutzer einerseits Experten auf ihrem Gebiet, aber andererseits vielleicht

Anfänger im Umgang mit Computern. Für diesen Anwenderkreis sind Programme die eine direkte Manipulation der Daten erlauben am besten geeignet.

4.4.3 Beachtung der Verschiedenheit der Benutzer

Bei der Entwicklung benutzerfreundlicher Systeme ist sehr stark darauf zu achten, daß geistige und körperliche Unterschiede der zukünftigen Anwender nicht unterschätzt werden [Shneider92]. Es gibt mehrere Möglichkeiten Benutzer in verschiedene Gruppen aufzuteilen und dann zu analysieren, ob das Benutzerinterface all diesen Gruppen genügt.

Nach Carl Jung existiert eine Theorie, die die Menschen in 4 zueinander entgegengesetzte Typen unterscheidet:

- *extrovertiert - introvertiert*: Extrovertierte lenken ihre Aufmerksamkeit auf äußere Reize und brauchen die Interaktion mit ihrer Umwelt. Introvertierte dagegen bevorzugen Strukturen an die sie gewöhnt sind, beziehen sich auf ihre eigenen Ideen und arbeiten allein und inhaltsbezogen.
- *gefühlsbezogen - intuitiv*: Gefühlsbezogene Typen werden von feststehenden Routinen angesprochen, sind gut auf dem Gebiet der Präzisionsarbeit und wenden gerne bekannte oder erlernte Fähigkeiten an. Intuitive Menschen lösen gerne neue Probleme und erforschen neue Zusammenhänge, aber wollen sich keine Zeit für Genauigkeit nehmen.
- *einfühlend - bewertend*: Einfühlsame Personen lernen gerne über neue Situationen, aber haben Probleme Entscheidungen zu treffen. Bewertende arbeiten nach einem genauen Plan, und versuchen den Plan auch einzuhalten wenn neue Fakten das Ziel verschoben haben.
- *fühlend - denkend*: Fühlende Typen wissen über die Gefühle anderer Bescheid, versuchen andere zufriedenzustellen und haben gute Beziehungen zu den meisten Personen ihrer Umgebung. Denkende Typen sind unemotionell, verhalten sich anderen Menschen gegenüber unpersönlich, und bringen Dinge gerne in eine logische Ordnung.

Eine andere Möglichkeit, die auf dem unterschiedlichen Wissen von Benutzern aufbaut ist die Einteilung in 3 Gruppen von [Shneider92]:

- *Anfänger*: Sie haben kein Wissen wie das System funktioniert und wie es zu bedienen ist. Auch haben sie meist nur wenig bis gar keine Erfahrung im Umgang mit Computern.

- *durchschnittliche Benutzer:* Können meist schon auf einige Erfahrungen mit verschiedenen Computersystemen zurückgreifen. Sie haben wenig bis keine Probleme die üblicherweise anfallenden Aufgaben zu lösen.
- *Experten:* Arbeiten schon viel und lange mit Computern und haben auch keine Schwierigkeiten beim Lösen komplexer Probleme. Sie erledigen ihre Aufgaben sehr schnell und erwarten vom System, daß es schnell und exakt auf ihre Eingaben reagiert. Umständliche oder immer wiederkehrende Operationen, die ihre Arbeitsgeschwindigkeit bremsen, werden als unangenehm empfunden.

Einige Beispiele wie bei verschiedenen Benutzergruppen sehr unterschiedliche Probleme bei der Nutzung von graphischen Betriebssystemen auftreten [Micro95]:

- *Anfänger haben sehr oft Schwierigkeiten im Umgang mit der Maus:* Beim Verschieben oder Doppelklicken von Objekten brauchen sie Zeit um eine gewisse Fingerfertigkeit zu entwickeln. Beim Verschieben ist es für sie schwierig, eine Maustaste gedrückt zu halten und gleichzeitig die Maus auf ein Ziel hinzubewegen. Ein Doppelklick mit der Maus ist nicht das gleiche wie zwei einzelne Klicks: Anfänger müssen erst lernen wieviel Zeit zwischen zwei Klicks vergehen darf, damit es ein Doppelklick ist. Sie glauben dann auch oft, daß immer, wenn sie etwas ausführen wollen ein Doppelklick notwendig ist.
- *Anfänger haben Probleme mit der Verwaltung von Fenstern am Bildschirm:* Sie erkennen nicht, daß überlappende Fenster eine dreidimensionale Darstellung sind. Daher kommt es immer wieder vor, daß Fenster, die von anderen überlagert wurden, von einem Anfänger nicht mehr gefunden werden.
- *Anfänger verstehen nicht die Dateihierarchie auf einem Speichermedium:* Sie haben auch Verständnisschwierigkeiten beim Verschieben und Kopieren von Daten.
- *Experten wollen effiziente Programme:* Sie wollen sich keine Erklärungen für weniger erfahrene Benutzer anschauen müssen, sondern möglichst schnell mit dem System interagieren.
- *Internationalisierung und Behinderungen:* Um ein System oder Programm für möglichst viele Personen nutzbar zu machen, muß man auch Rücksicht auf internationale Nutzung oder behinderte Menschen Rücksicht nehmen.

4.4.4 Hauptinteraktionsstile beim Umgang mit Computern

Nachdem die Aufgaben, die ein System zu erfüllen hat, festgelegt wurden, muß sich der oder die Designer des Benutzerinterfaces überlegen, welche Interaktionsstile dem Anwender angeboten werden sollen [Shneider92] [Preece94] [Foley90]:

- *Menuauswahl:* In Menuauswahlsystemen werden dem Anwender eine Reihe von Möglichkeiten angeboten und er kann sich das Passende auswählen. Wenn die Terminologie und die Bedeutung der einzelnen Menüpunkte verständlich und leicht voneinander unterscheidbar ist, ist die Einlernzeit für den Benutzer sehr gering. Der große Vorteil dieses Stiles liegt in der klaren Struktur des Entscheidens, da immer nur wenige Möglichkeiten angeboten werden. Er bietet auch die Möglichkeit eine Online-Hilfe und Fehlermeldungen einfach zu integrieren. Designer von Menusystemen müssen sehr sorgfältig die anfallenden Aufgaben abschätzen, damit für alle Funktionen entsprechende Menüpunkte existieren. Die Nachteile der Menuauswahl liegen in der Gefahr von zu vielen Menus, Einbremsung von schnell arbeitenden Experten und daß die Darstellung des Menus am Bildschirm sehr viel Platz erfordert. Dieser Stil ist sehr gut für Anfänger und Fortgeschrittene geeignet, aber für Experten kann er durchaus zu langsam sein. Abhilfe dagegen schaffen sogenannte „Shortcuts“, die die Möglichkeit bieten durch Tastenkombinationen direkt einzelne Menüpunkte aufzurufen.
- *Formular:* Wenn die Eingabe einer großen Anzahl von Daten erforderlich ist, kann dem Benutzer eine Anzahl von Feldern angeboten werden in die er relativ einfach seine Werte eintippen kann. Er hat auch die Möglichkeit mit der Maus oder mit den Cursorstasten von einem Eingabefeld zum nächsten zu springen. Es wird also ein „gedrucktes Formular auf Papier“ am Bildschirm nachgebildet. Die Vorteile liegen in der Vereinfachung der Dateneingabe und einer geringen Einschulungszeit. Die Nachteile sind, daß der Anwender wissen muß, wo er welche Daten eingeben kann, in der Größe des Formulars, welches meist den ganzen Bildschirm bedeckt und darin, daß der eintippenden Person die Initiative entzogen wurde. Dieser Stil ist gut für Benutzer geeignet, die oft viele Daten eingeben müssen.
- *Kommandosprache:* Die einzelnen Befehle werden direkt eingetippt und auch sofort vom System ausgeführt. Vorteile: sehr flexibel, schnell zu bedienen, überläßt dem Benutzer die Initiative: mehrere Kommandos können mit Macros zu einem neuen zusammengefaßt werden. Nachteile: Online-Hilfe und Fehlerbehandlung schwer zu implementieren, große Einlernzeit, Kommandos sind meist so komplex, daß sie bald wieder vergessen werden. Dieser Interaktionsstil ist sehr gut für Experten geeignet, da sie damit schnell arbeiten können und auch über das entsprechende Wissen über die einzelnen Kommandos verfügen.
- *natürliche Sprache:* In einfachen Sätzen wird beschrieben, welche Aktionen der Computer ausführen soll. Wird momentan noch sehr selten verwendet, da es noch keine Systeme gibt, die die Komplexität der menschlichen Sprache beherrschen. Der Vorteil wäre es den Anwendern eine Einlernphase, die bei allen anderen Interaktionsstilen notwendig sind, zu ersparen. Nachteile liegen darin, daß der

Computer immer nachfragen muß, ob er die Eingabe richtig interpretiert hat und das viel Text eingegeben werden muß um eine gewünschte Aufgabe zu umschreiben.

- *direkte Manipulation:* Der Anwender kann direkt mit Objekten arbeiten, er sieht immer genau wie sich seine Befehle auf die Objekte auswirken. Heute verwenden fast alle Systeme diesen Stil. Die Vorteile sind eine visuelle Präsentation der Daten, die geringe Einlernzeit, die hohe Merkbarekeit über die Zeit, die Möglichkeit Fehler zu verhindern, die hohe subjektive Zufriedenheit, dem Gefühl das System zu beherrschen und die Möglichkeit mit dem System zu experimentieren. Die Nachteile bestehen aus dem großen Aufwand diesen Stil zu programmieren, in der Notwendigkeit schneller graphischer Darstellungsmöglichkeiten und in Schwierigkeiten bei Personen die visuelle Probleme haben.

Allgemeine Designprinzipien nach [Apple87], [Micro95], [Foley90]:

- *Metapher aus der realen Welt:* Verwendung von konkreter und klarer Metapher, sodaß Anwender gewisse Erfahrungen der realen Welt in einem System anwenden können. Wenn die Möglichkeit besteht, sollen diese Metapher durch visuelle Effekte und Audio unterstützt werden. Viele Menschen verwenden heute Computer, haben aber keine jahrelange Erfahrung damit. Sie kennen sich nur in der realen Welt aus, haben dort aber einen großen Erfahrungsschatz. Mit Hilfe von Metapher kann man diesen Erfahrungsschatz für eine einfachere Nutzung des Computers verwenden.
- *Direkte Manipulation:* Die Anwender möchten das Gefühl haben, daß sie genau wissen welche Aktivitäten der Computer ausführt. Personen erwarten von ihren physikalischen Aktionen physikalische Ergebnisse. Wenn z.B. eine Taste auf der Tastatur gedrückt wird, dann soll ein Geräusch ertönen und gleichzeitig das Zeichen am Bildschirm erscheinen. Wenn in einem Zeichenprogramm eine Linie gezeichnet werden soll, dann muß diese Linie sofort sichtbar werden und sich auch verändern lassen. Benutzer möchten zu jedem Zeitpunkt sehen, welche Möglichkeiten ihnen offen stehen, sie möchten gewarnt werden bevor durch eine Unachtsamkeit Daten zerstört werden. Wenn ein Fehler auftritt soll er den Grund beinhalten und auch Alternativen anbieten.
- *Sehen und Zeigen anstatt Erinnern und Eintippen:* Anwendern soll die Möglichkeit geboten werden aus am Bildschirm angebotenen Alternativen zu wählen, d.h. sie sollen etwas sehen und wiedererkennen anstatt sich erst mühsam zu erinnern. In einem guten Benutzerinterface soll man sich an nichts erinnern müssen, sondern das System soll alle Möglichkeiten anbieten. Es ist zu beachten, daß sehr viele Programmierer keine Probleme haben mit einer Kommandosprache zu arbeiten, die auf der Basis von Erinnerungen und boolescher Logik arbeitet. Allerdings sind die meisten Benutzer eines Computers keine Programmierer.

- *Konsistenz*: Gute Anwendungen sind in sich selbst und mit anderen Programmen konsistent. In Microsoft Windows sind z.B. sehr viele Funktionen bei allen Programmen gleich. Durch Konsistenz erhält man das Gefühl, daß die Anwendung vertraut und ihr Verhalten vorhersagbar ist. Konsistenz in einem System bedeutet Konsistenz in allen Produkten, im Betriebssystem und in den verwendeten Metaphern.
- *WYSIWYG (What You See Is What You Get)*: Der Anwender soll jederzeit und sofort sehen können, wie sich seine Befehle auf seine Daten auswirken. Es sollen z.B. keine Unterschiede zwischen einem am Bildschirm dargestellten Brief und seinem auf Papier ausgedrucktem Gegenstück sein.
- *Anwender kontrolliert das System*: Die Aktion soll vom Benutzer ausgehen und nicht, daß er auf das System reagieren muß. Für ihn ist es ein sehr positives Gefühl zu erkennen, daß er die Kontrolle über den Computer hat und nicht umgekehrt. Er würde sich sehr bald eingeschränkt fühlen, wenn der Computer ihm jeden Schritt genau vorschreibt der ausgeführt werden muß. Eine Ausnahme sind Assistenten, die ihn durch eine sehr komplexe Aufgabenstellung leiten.
- *Rückmeldungen des Systems*: Der Anwender möchte immer genau über den aktuellen Status des Systems informiert sein. Daher ist es notwendig, daß für jede getätigte Aktion vom Computer eine kurze Rückmeldung erfolgt. Vor allem bei länger andauernden Rechenoperationen ist eine ständige Rückmeldung unbedingt notwendig.
- *Wiederherstellen eines älteren Status*: Da Benutzer immer wieder Fehler begehen, sollte ihnen eine einfache Möglichkeit geboten werden, den Fehler rückgängig zu machen. Generell sollten alle Aktionen wieder rückgängig gemacht werden können um damit den Benutzer zum Experimentieren einzuladen, d.h. er kann nach dem Prinzip „Versuch und Irrtum“ lernen. Alle Aktionen die unumkehrbar sind, müssen dem Anwender immer vorher mitgeteilt werden.
- *Stabilität*: Personen fühlen sich nur in Programmen wohl die immer zuverlässig, verständlich und stabil arbeiten.
- *ästhetische Integrität*: Schlechte visuelle Darstellung oder ein Design das nur beeindruckt und nicht informiert verringern die Effektivität der Mensch-Maschine-Kommunikation.
- *Einfachheit*: Ein Benutzerinterface sollte immer so einfach wie möglich zum Lernen und zum Verwenden sein. Es muß außerdem den Zugang zur gesamten Funktionalität eines Systems bieten.

4.4.5 Regeln für das Design von Benutzerinterfaces

Regeln für das visuelle Design:

Was wir sehen, beeinflußt wie wir fühlen und was wir verstehen [Micro95], [Foley90]. Visuelle Information kommuniziert nonverbal, aber sehr effizient. Sie kann motivieren, leiten oder abstoßen. Effektives visuelles Design leistet viel mehr als nur Dekoration. Es ist ein sehr wichtiges Kommunikationsmittel. Wie Information am Bildschirm angeboten wird, kann einen großen Unterschied bewirken: der Benutzer ist informiert oder er ist verwirrt oder überlastet. Auch die beste Funktionalität eines Produktes leidet unter einer schlechten visuellen Präsentation. Von einem Bildschirm wird auf die gleiche Weise wie z.B. von einer Zeitung oder einem Plakat gelesen: Das Auge wandert zuerst zu den bunten Elementen und dann zu den schwarzen und weißen, zuerst zu isolierten dann zu gruppierten Elementen und dann zu Graphiken und dann erst zu Text.

Allgemeine Regeln für das Design eines Layouts:

- *Hierarchie der Information:* Die Platzierung der Information erfolgt nach ihrer relativen Wichtigkeit. Diese Ordnung beeinflußt welche Elemente ein Benutzer zuerst betrachtet bzw. betrachten soll. Hier sollten immer drei Fragen gestellt werden:
 - Welche Information ist für den Benutzer am wichtigsten?
 - Was möchte der Benutzer als erstes, zweites, drittes ausführen?
 - Was sollte er als erstes, zweites, drittes am Bildschirm betrachten?
- *Zentrum und Betonung:* Dienen zur Definition von Prioritäten. Die Festlegung des Zentrums bedeutet die zentrale Idee zu identifizieren. Wichtige Elemente müssen betont werden um sie von anderen zu isolieren. In internationaler Software ist dies schwierig, da z.B. Europäer die linken Bereich zuerst erfassen, Araber (deren Schrift von rechts nach links geschrieben wird) bevorzugen den rechten Bereich.
- *Struktur und Balance:* Am Bildschirm muß immer eine Struktur der angebotenen Information erkennbar sein. Genauso müssen die Elemente ausgewogen auf der Fläche verteilt werden.
- *Beziehungen zwischen Elementen:* Elemente, die in direktem Zusammenhang zueinander stehen, sollen auch räumlich nahe beisammen liegen um den Benutzer beim Erkennen eines Zusammenhanges zu unterstützen.
- *Lesbarkeit und Lesefluß:* Bei der Entwicklung eines Layouts für den Bildschirminhalt stellen sich folgende Fragen:
 - Kann die Idee oder das Konzept in einer einfacheren Form präsentiert werden?

- Kann der Benutzer einfach durch die angebotene Information wandern?
- Haben alle Elemente die sich am Schirm befinden, einen Grund für ihre Anwesenheit?
- *Einheit und Integration:*
 - Wie verhält sich ein neu entwickeltes Design zu seiner Umgebung?
 - Wie verhält sich das Design gegenüber anderen Programmen, die bereits in Verwendung sind?
 - Wird es in das Design des übergeordneten Systemes integriert?
 - Wie arbeiten die verschiedenen Elemente, die am Schirm dargestellt werden, zusammen?

Allgemeine Regeln für das Arbeiten mit Farben [Micro95], [Shneider92], [Heller89]:

Farben sind eine sehr wichtige Eigenschaft eines visuellen Interfaces. Wenn Farbe unkritisch benutzt wird, kann es einen negativen oder abschreckenden Effekt bewirken.

Bei der Verwendung von Farben sind einige Punkte zu beachten:

- Aufzeigen von Beziehungen oder Gruppierungen. Menschen glauben oft, daß Objekte mit gleicher Farbe auch eine Beziehung zu einander haben. Allerdings glauben sie genausooft, daß die „Farben“ schwarz und weiß zusammengehören.
- Setzen von Akzenten in einem uninteressantem Bildschirminhalt.
- Unterstützung der logischen Organisation eines Layouts.
- Erregung der Aufmerksamkeit des Anwenders, z.B. Fehlermeldungen in roter Farbe bewirken mehr als in grauer Farbe.
- Die Farbkodierung sollte im gesamten System immer konsistent sein.
- Die Verwendung einer Farbe für eine spezielle Bedeutung (z.B. rot bedeutet Liebe) kann bei internationaler Software durchaus zu Problemen führen (die Farbe der Trauer ist in Mitteleuropa schwarz, in Japan hingegen weiß).
- Das gleiche Problem kann auch schon bei nationaler Software problematische Nebeneffekte haben:
 - für Personen, die sich viel mit Autos beschäftigen bedeutet rot Stop, gelb Vorsicht und grün losfahren.
 - für einen Wirtschaftsexperten bedeutet rot einen Verlust und schwarz einen Gewinn.
 - für Chemiker und Installateure ist rot heiß und blau kalt.

- für Geographen ist blau Wasser, grün Wald bzw. Wiese und gelb bedeutet Wüste.
- Farbe hat sehr subjektive Eigenschaften. Jeder hat einen anderen Farbgeschmack. Wenn jemand von einer Farbe begeistert ist, findet sich sicher bald jemand anders, der diese Farbe als abstoßend empfindet.
- Ein Teil der Endanwender arbeitet nur mit monochromen Bildschirmen und hat dadurch Schwierigkeiten, Information die nur durch unterschiedliche Farben dargestellt wird, zu erkennen. Das gleiche gilt für all jene die farbenblind sind. Forschungen haben ergeben, daß ca. 9 Prozent aller Menschen Probleme beim Erkennen von Farben haben.
- Farbe kann daher nur als zusätzliche, sekundäre Möglichkeit der Information verwendet werden. Es wird empfohlen, ein Design zuerst in schwarz und weiß zu gestalten und dann erst Farbe als zusätzliche Informationsquelle zu benutzen.
- Die Verwendung von Farben bei einer Statusänderung kann vorteilhaft sein, z.B. ein Wert gelangt in einen kritischen Bereich und wird dann rot dargestellt.
- Ausdrucken von Farbdarstellungen auf Papier kann problematisch sein, bzw. können die Farben am Papier sich von denen am Bildschirm unterscheiden.
- Verwendung einer limitierten Anzahl von Farben gleichzeitig: Obwohl das menschliche Auge Millionen von Farben unterscheiden kann, können zuviele Farben in einem Layout den Benutzer verwirren und ihn überfordern die eigentliche Information zu erkennen.
- Gedämpfte, komplementäre Farben sind meist besser als kräftige, helle Farben.
- Die Hintergrundfarbe sollte sehr neutral sein, am besten ein helles grau. Komplementäre Farben als Vorder- und Hintergrund zu verwenden bewirkt nur, daß das menschliche Auge Schwierigkeiten bei der Fokussierung hat. Dunkle Farben erscheinen weiter entfernt als hellere.
- Es soll die Möglichkeit geboten werden, die Farben der Benutzeroberfläche veränderbar zu gestalten damit sich jeder einzelne Benutzer jene Farben auswählen kann die er als angenehm empfindet.

Regeln für das Dialogdesign:

In [Shneider92] werden 8 „goldene Regeln“ für Dialogdesign empfohlen:

- *Bemühung um Konsistenz:* Diese Regel wird am meisten mißachtet und ist aber gleichzeitig am einfachsten zu erfüllen: In ähnlichen Situationen sollen konsistente Abläufe von Aktionen erfolgen, identische Terminologie in Menutexten,

Dialogtexten und in der Online-Hilfe, Abweichungen von der Konsistenz sollen nur in Ausnahmefällen geschehen.

- *Möglichkeit von Shortcuts für Experten:* Funktionen und Befehle, die oft benötigt werden, können entweder über ein Menu oder schneller durch Drücken einer Funktionstaste aufgerufen werden.
- *Angebot informativer Rückmeldungen:* Für jede Aktion des Benutzers sollte eine Antwort des Systems erfolgen. Dies ist besonders bei längeren Rechenzeiten notwendig.
- *Zusammenhänge in Dialogen kennzeichnen:* Abläufe von Aktionen sollen in einen Anfang, in einen Mittelteil und in ein Ende unterteilt werden.
- *Einfache Fehlerbehandlung:* Das System sollte so designed werden, daß dem Benutzer schwerwiegende Fehler nicht möglich sind. Falls trotzdem ein Fehler auftritt, soll das System ihn erkennen und einen einfachen und verständlichen Lösungsweg anbieten, bzw. eine Möglichkeit anbieten, den Fehler rückgängig zu machen.
- *Einfaches Zurücknehmen von Aktionen:* Wenn möglich, sollte jede Aktion des Benutzers durch ihn selbst wieder rückgängig gemacht werden können. Dies soll Anwender in der Erforschung einer neuen Lösungsmöglichkeit für ihr Problem unterstützen.
- *Anwender agiert - Computer reagiert:* Menschen fühlen sich unwohl, wenn ihnen der Computer vorschreibt, welche Aktion sie als nächstes tätigen müssen. Benutzer sollen das Gefühl haben, die Kontrolle über den Computer zu besitzen und nicht umgekehrt.
- *Anwender soll sich nichts merken müssen:* Da das menschliche Gehirn nur wenige Informationen gleichzeitig im Arbeitsgedächtnis halten kann, muß der Bildschirminhalt einfach und informativ sein. Wenn z.B. eine Datei geladen werden soll, dann müssen immer alle möglichen Dateien zur Auswahl angeboten werden anstatt nur nach einem Dateinamen zu fragen.

Regeln für Datendisplays [Shneider92]:

- Bei jedem Schritt in einer Eingabesequenz sollen immer jene Daten am Schirm angeboten werden, die der Benutzer für die Eingabe benötigen könnte.
- Daten für die Anzeige sollen in einer direkten und verwendbaren Form dargestellt werden. Der Anwender soll keine Konvertierungen vornehmen müssen.
- Für jeden einzelnen Datentyp soll ein konsistentes Layout während des gesamten Programmablaufes zur Verfügung gestellt werden.

- Verwendung von kurzen, einfachen Sätzen.
- Die Sätze sollen positiv formuliert werden.
- Verwendung logischer Prinzipien bei der Sortierung von Listenelementen. Wenn kein Prinzip anwendbar ist, dann soll die Liste alphabetisch sortiert werden.
- Die Bezeichnungen von Eingabefeldern sollen nahe genug beieinanderliegen um sie als zusammengehörig zu erkennen. Der Mindestabstand sollte aber ein Leerzeichen betragen.
- Spalten die linkszentriert sind ermöglichen ein schnelles lesen oder suchen nach einem bestimmten Element.
- Bei Bildschirmhalten, die auf mehrere Seiten verteilt sind, soll jede Seite bezeichnet werden, um die Beziehungen zwischen den Seiten erkennbar zu machen.
- Jede Darstellung sollte mit einem Titel oder mit einer Kopfzeile beginnen um kurz den Inhalt oder die Intention dieser Anzeige zu erklären. Es sollte sich mindestens eine Leerzeile zwischen dem Titel und dem Rest der Darstellung sein.
- Bei Größenkodierungen sollte das größere Symbol mindestens eineinhalb mal größer sein als das kleinere.
- Bei blinkender Information, z.B. blinkender Text, sollte die Blinkrate zwischen 2 und 5 Herz liegen mit einem sichtbaren Intervall von mindestens 50% der Gesamtzeit.
- Bei großen Tabellen sollen für den Anwender die Spaltenüberschriften immer sichtbar sein.

Regeln für Dateneingabe und Formulare [Shneider92]:

Die Dateneingabe erfordert meist einen großen Teil der Arbeit eines Anwenders. Das Potential für Frustration und Eingabefehlern ist daher entsprechend hoch.

Grundregeln für Dateneingabe:

- *Konsistenz von Dateneingabeaktionen:* Gleiche Sequenzen von Aktionen sollten unter allen Umständen beibehalten werden.
- *Minimale Eingabeaktionen vom Benutzer:* Je weniger Tasten der Benutzer drücken muß, umso effizienter kann er seine Daten eingeben. Redundante Eingabe von Daten sollte vermieden werden.
- *Anwender soll sich möglichst wenig merken:* Er soll sich an keine speziellen Codes oder Kommandos mit komplexer Syntax erinnern müssen.
- *Kompatibilität zwischen Dateneingabe und Datenanzeige:* Das Format der Dateneingabe muß gleich dem Format der Datenanzeige sein.

- *Flexibilität bei der Eingabe:* Der Benutzer soll die Reihenfolge in der er die Daten eingeben will selbst festlegen können.

Designregeln bei Eingabefeldern [Shneider92]:

- entsprechender Titel für das Formular, keine Fachbezeichnungen die nur für Experten verständlich sind
- verständliche Instruktionen zur Nutzung des Formulars
- logische Gruppierungen und Aneinanderreihungen von zusammengehörigen Elementen
- visuell ansprechendes Layout
- allgemein verständliche Bezeichnungen für jedes Eingabefeld
- konsistente Terminologie und Abkürzungen innerhalb des gesamten Formulars
- genügend Raum oder Trennlinien zwischen den einzelnen Feldern
- einfache und verständliche Bewegungsmöglichkeiten für den Cursor
- Fehlerkorrektur für einzelne Buchstaben und gesamte Felder
- Fehlermeldungen bei unakzeptablen Eingabewerten
- Eingabefelder, welche nur optional sind, müssen deutlich gekennzeichnet werden

Regeln für Assistenten (wizards):

Ein Assistent ist eine spezielle Form von Benutzerhilfe der die Automatisierung eines Ablaufes bewirkt [Micro95].

Assistenten helfen bei der Durchführung einer Aufgabe die komplex ist und viel Erfahrung verlangt. Sie werden meist für selten verwendete oder unübersichtliche Aufgaben verwendet, bei denen der Benutzer Schwierigkeiten hat sie zu lernen oder auszuführen. Assistenten sind aber nicht dafür geeignet, einem Anwender zu lernen wie ein Problem gelöst wird. Sie werden so entworfen, daß viele Schritte und auch die Komplexität eines Problems vor dem Benutzer versteckt werden. Es ist zu beachten, daß ein Assistent keine Lösung für ein ansonsten schlechtes Design darstellt.

Ein Assistent besteht aus mehreren Seiten die verschiedene Eingaben vom Anwender fordern. Dieser kann durch die einzelnen Seiten navigieren. Auf der ersten Seite sollte eine kurze Beschreibung der Intention des Assistenten erfolgen. Alle weiteren Seiten sollen leicht verständlich designed werden. Es ist meist besser, eine größere Anzahl von Seiten zu haben, als wenige, die dafür unübersichtlich sind.

Der Anwender sollte immer das Gefühl haben, er habe die Kontrolle über den Prozeß, der gerade ausgeführt wird.

Richtlinien für Texte in Assistenten:

- Verwendung von Wörtern, die den Benutzer persönlich ansprechen (z.B. „Sie“, „Ihre...“)
- Verwendung von Fragestellungen anstatt Befehlen („Welche Option möchten sie nutzen?“) statt „Wählen sie eine Option!“)
- kurze, allgemein bekannte Wörter
- Vermeidung technischer Terminologie die Anfänger möglicherweise verwirren kann
- möglichst kompakt formulierte Sätze
- kurze, klare und verständliche Schreibweise

4.5 Designmethodik

Effektives Interfacedesign ist mehr als das Verfolgen einiger Regeln. Es benötigt eine benutzerbezogene Designmethodik. Weiters involviert es eine frühe Planung des Interfaces und eine kontinuierliche Weiterentwicklung während des gesamten Entwicklungsprozesses der Software [Micro95].

Eine wichtige Entscheidung im Design eines Produktes ist die Zusammenstellung eines Teams, welches das Design entwickelt und die Programmierung dafür durchführt. Es ist wichtig, immer ein Gleichgewicht zwischen den verschiedenen Disziplinen und den Fähigkeiten einzelner zu schaffen.

4.5.1 Ein effektives benutzerbezogenes Design beinhaltet mehrere einzelne Phasen, den sogenannten Designzyklus:

- *Design*: Der Beginn der Arbeit an einem Design ist der kritischste Punkt, da hier die Entscheidung über das generelle Aussehen eines Produktes gefällt wird. Fehler, die hier entstehen, lassen sich später nur mehr sehr schwierig korrigieren. Dieser Teil des Zyklus beinhaltet nicht nur das Definieren des sachlichen Bereiches und der Features eines Produktes sondern auch das Verständnis wer die Anwender sind, und welche Aufgaben, Vorstellungen und Ziele sie verwirklichen wollen. Es erfolgt eine Definition eines konzeptuellen Rahmens um das System dem Wissen und der Erfahrung der zukünftigen Anwender anzupassen.
- *Prototyping*: Nachdem die Designphase abgeschlossen wurde, erfolgt die Erstellung einiger Prototypen die die grundsätzlichen Aspekte des neuen Interfaces beinhalten. Dies kann einerseits mit Papier und Bleistift oder mit speziellen „prototyping tools“ geschehen. Diese Prototypen bilden die Basis für Diskussionen und Weiterentwicklungen am Interface.

- *Test*: Benutzerbezogenes Design beinhaltet die Einbeziehung des Endanwenders in die Entwicklung. Das Testen einiger Teile oder des gesamten Interfaces gibt Aufschluß über die Qualität des Designs. Die Tests geben Auskunft über das Empfinden des Anwenders, über seine subjektive Zufriedenheit, und über Fragen und Probleme die die Testperson bei der Anwendung des Interfaces aufwirft. Es ist wichtig, daß die Tester keine Kollegen oder andere Softwareentwickler sondern nur Personen aus der tatsächlichen Zielgruppe des Programmes sind.
- *Iteration*: Da das Testen oft Programmier- und Designfehler erkennen läßt, erfolgt nach der Testphase eine erneute Designphase. In der Praxis werden diese 3 Phasen (Design, Prototyping, Test) sehr oft durchlaufen bis ein Produkt seine Marktreife erlangt hat.

Dieser gesamte Designzyklus sollte parallel mit der Entwicklung der Funktionalität des eigentlichen Programmes ablaufen, da der Beginn eines Interfacedesigns erst nach Fertigstellung aller anderen Programmteile meist keine Zeit mehr für ein gutes Design läßt. Erfahrungsgemäß dauert die Entwicklung eines guten Interfaces genauso lange wie die Entwicklung der eigentlichen Funktionalität eines Programmes.

Das Testen der Verwendbarkeit eines Interfaces ist entscheidend für den Designprozeß, aber nur das Testen von Prototypen ist zu wenig. Die Einschätzung der Nutzbarkeit sollte gleichzeitig mit den ersten Schritten in der Produktentwicklung erfolgen. Damit lassen sich sofort Ergebnisse erzielen die in den weiteren Designprozeß einfließen. Es ist zu beachten, daß die Erfahrungen der Benutzer unbedingt beachtet werden muß [Mirco95].

4.5.2 Möglichkeiten zum Testen der Verwendbarkeit eines Produktes (usability testing)

Usability Tests beinhalten einen großen Bereich von Techniken und Ressourcen, wie Spezialisten, schallisolierte Räume, Einwegspiegel und Aufnahmeggeräte. Aber schon auch ein Büro- oder Konferenzraum, Kassettenrekorder, Stoppuhr und Notizblock reichen für kleinere Tests aus. Typischerweise genügen kleine, wohlformulierte Tests mit nur 6-10 Teilnehmern um 80 bis 90 Prozent aller Designfehler zu erkennen.

Wie der Designprozeß selbst, beginnen Usability Tests mit dem Definieren der Testpersonen und den Testzielen. Bei der Entwicklung eines Tests ist auf die grundsätzliche Problemstellung und nicht auf das Testen von Features zu achten. Auch wenn nur sehr spezielle Dinge getestet werden sollen, der Endanwender wird diese immer im Zusammenhang mit dem gesamten System nutzen. Es ist immer sinnvoll einen Probedurchlauf des Tests mit einer Person aus der Testgruppe durchzuführen um zu erkennen ob die Szenarios, der Prototype und die Aufnahmeggeräte korrekt arbeiten und sinnvolle Ergebnisse liefern können.

Bei der Ausführung des Tests soll eine angenehme Atmosphäre für die Testperson geschaffen werden: ein stiller Raum, mit einem entsprechenden Schild vor der Tür um Störungen zu vermeiden. Für die Testperson soll es bequem, freundlich und komfortabel sein. Sie muß wissen, daß nicht sie sondern das Produkt getestet wird.

Während des Testes soll genügend Zeit vorhanden sein, um alle Aufgaben lösen zu können. Die Testperson soll in ihrer Arbeit nicht unterbrochen werden, wenn Probleme auftreten, nur vorsichtig und eher mit allgemeinen Antworten helfen. Dem Tester soll die Möglichkeit geboten werden, jederzeit die auszuführenden Aufgaben zu beenden.

Um die Tests auswerten zu können ist es sehr hilfreich die Person zu bitten während der Arbeit laut zu denken. Durch eine gleichzeitige Aufnahme mit einem Tonbandgerät oder einer Videokamera ist es dann bei der Auswertung leicht möglich die Gedankengänge des Testers nachzuvollziehen.

Die Testergebnisse werden abschließend ausgewertet und allen Mitarbeitern des Designteams zur Verfügung gestellt um auf deren Basis die aufgetretenen Probleme bei der Nutzung des Produktes zu beheben.

5 SUCHEN NACH INFORMATION (INFORMATION RETRIEVAL)

5.1 Was ist Information Retrieval (IR)?

Unter Information Retrieval wird im weitesten Sinne des Wortes jede Art der Wiedergewinnung (maschinell) gespeicherter Daten verstanden [Mresse 84], [Lancaster 78], [Salton 68].

Die Entwicklung dieses Wissenschaftszweiges der Informatik liegt an der zunehmenden Flut von Informationen in Form von Bücher, Zeitschriften und elektronischen Medien. Ein weiterer Grund ist die zunehmende Automatisierung in Büros durch die Anwendung von kleinen, leistungsfähigen Computern im Zusammenhang mit den Möglichkeiten die das Internet bietet.

Der Begriff Information Retrieval sagt aber noch nichts über die Art der Wiedergewinnung der Information aus. Erst seit einigen Jahren gibt es IR-Systeme die einen interaktiven Dialog mit dem Benutzer erlauben. Davor wurden Anfragen an das Informationssystem nicht persönlich sondern nur über einen Experten (den Operator) durchgeführt und es konnte durchaus ein paar Tage dauern, bis die Antwort geliefert wurde. Diese Arbeitsweise hatte aber den Vorteil, daß der Experte genauestens mit dem System vertraut war und daher relativ rasch relevante Informationen erhielt. Bei den heutigen Systemen ist es meist genau umgekehrt [Raitt 84]: Jeder darf beim IR-System anfragen, erhält recht schnell eine Antwort, da er aber nicht oft damit arbeitet, muß er aufgrund mangelnder Erfahrung viel öfter Anfragen stellen, bis er alle für ihn relevanten Informationen erhält.

Weiters wird beim IR-Begriff keine Aussage über die Art der Speicherung der Daten getroffen. Diese können in einer speziellen Datenbank oder in einzelnen Dateien gespeichert werden. Im Fall der Datenbank ist das IR-System nur das Bindeglied zwischen der Datenbank und dem Benutzer, im zweiten Fall der Dateien ist es auch noch für die Speicherung und Dateiverwaltung zuständig [Houghton 84].

5.2 Grundidee

Die Grundidee von Information Retrieval beruht auf der Vorstellung eines Karteisystemes, in welchem sich die einzelnen Informationseinheiten auf Karteikarten befinden [Mresse 84]. Die Suche bezieht sich daher immer auf den Inhalt bestimmter Karteikarten. Das Ziel des Information Retrieval ist es diese Karteikarten bzw. deren

Inhalt maschinell bearbeiten zu können, d.h. Daten eingeben, verändern, löschen und vor allem wiederfinden und übersichtlich dem Benutzer präsentieren. Die manuellen Methoden zur Arbeit mit Karteikarten sind heute schon sehr ausgereift: Existierende Systeme sind in der Lage eine begrenzte Anzahl von Karteikarten mit einer begrenzten Anzahl von Merkmalen zu bearbeiten. Trotz alledem haben manuelle Systeme einige Schwächen die durch computerunterstützte Informationssysteme beseitigt werden können [Rowley 87]:

- Die auf einer Karteikarte speicherbare Informationsmenge ist immer begrenzt.
- Die Anzahl der mit vertretbarem Aufwand zu verwaltenden Karteikarten ist limitiert.
- Die Merkmale, mit denen nach einzelnen Karteikarten gesucht werden kann, müssen im Voraus bestimmt werden und sind auch in ihrer Anzahl begrenzt.
- Die Möglichkeit einzelne Suchmerkmale logisch miteinander zu verknüpfen ist entweder gar nicht oder nur beschränkt vorhanden.

Aber manuell verwaltete Karteikarten haben auch zwei Vorteile deren Eigenschaften nicht einfach auf computerunterstützten Systemen implementiert werden können [Rowley 87]:

- Die Karteikarten erfordern keine bestimmte Form der Eintragungen (es kann sich z.B. ohne Probleme jederzeit neben oder inmitten eines Textes eine Skizze befinden).
- Das Erstellen oder Ausfüllen von Karteikarten kann auch durch einen Laien erfolgen.

5.3 Begriffsdefinitionen

Um sich von der Vorstellung einer physikalischen Karteikarte als Datenträger zu lösen, kann man den Begriff der Informationseinheit definieren [Mresse 84]. Eine Informationseinheit umfaßt einen Text und seine dazugehörigen Deskriptoren. Ein Text ist üblicherweise eine unstrukturierte und unformatierte Folge von Zeichen, Wörtern oder Sätzen. Ein Deskriptor (Merkmal, Attribut oder Eigenschaft) ist eine Zeichenfolge bestehend aus einem oder mehrerer Wörter die auf ein besonderes Merkmal des Textes hinweist. Etwaige Regeln zur Darstellung der Informationseinheit werden meist unabhängig vom Text gespeichert und als Ein- bzw. Ausgabemasken bezeichnet. **Mögliche Daten die mit Hilfe eines Information-Retrieval-Systemes (IRS) gespeichert und wiedergefunden werden können sind z.B.:**

- *alle Daten über bestimmte Objekte* (Personal, Bücher, Patienten oder Krankheiten oder Kunstgegenstände)

- *Objekte können aber auch als Ganzes gespeichert werden* (die Korrespondenz in einem Sekretariat, Programme und deren Dokumentation, persönliche Notizen, Zeichnungen oder Graphiken)

Bevor aber diese Daten mit einem IRS bearbeitet werden können, müssen sie in eine für die Wiedergewinnung geeignete Form gebracht werden. Diese Umwandlung erfolgt meist in zwei Schritten:

- Daten aufnehmen
- und anschließend die Daten in das System laden.

Unter der Datenaufnahme sind alle Tätigkeiten zu verstehen, die mit dem Erstellen der einzelnen Informationseinheiten verbunden sind:

- *IRS*: Das Bereitstellen einer benutzerfreundlichen Methode zur Dateneingabe (siehe auch das Kapitel über Mensch-Maschine-Kommunikation).
- *IRS*: Das Bereitstellen einer Möglichkeit zur Definition der Struktur einer Informationseinheit.
- *Benutzer*: Die Aufnahme der Texte. Dabei erfolgt ein unterschiedliches Vorgehen, wenn der Text Informationen über ein Objekt, eine Referenz zu einem Objekt oder das Objekt selbst enthält.
- *Benutzer*: Die Aufnahme der Deskriptoren. In einigen Fällen erfolgt dies auch automatisch durch das IRS. Trotzdem kann es vorkommen, daß der Benutzer diese Deskriptoren noch manuell nachbearbeiten muß.

Das Resultat dieser Datenaufnahme sind Rohdaten welche aus einer Menge von Informationseinheiten bestehen. Beim Ladevorgang wird durch Umorganisation der Rohdaten, sowie durch Speicherung von Zusatzinformationen über die Struktur der Daten, eine Verbesserung der Zugriffsmöglichkeiten bei der Abfrage erreicht. Die Struktur der Daten, welche beim Ladevorgang entsteht, sowie die Beziehungen zwischen den Daten werden als Datenorganisation bezeichnet. Die Datenaufnahme und deren Ladevorgang können auf zwei unterschiedliche Arten behandelt werden:

- *getrennt*: Aufgenommene Daten werden als Rohdaten zwischengespeichert und erst zu einen späteren Zeitpunkt geladen.
- *gemeinsam*: Jede Informationseinheit wird sofort nach Beendigung der Datenaufnahme in das System geladen.

Der Begriff Datenabfrage umfaßt alle Tätigkeiten die der Wiedergewinnung der Informationen dienen.

5.4 Klassifizierung von Information-Retrieval-Systemen

Anhand der vorigen Begriffsdefinitionen besteht die Möglichkeit IRS in verschiedene Klassen mit bestimmten Eigenschaften zu unterteilen [Mresse 84]:

- *Nach Art der zur Verfügung stehenden Information:*
 - Die gesamten Texte stehen in elektronischer Form zur Verfügung (Volltext-IRS).
 - Kürzungen oder Zusammenfassungen der Texte stehen in elektronischer Form zur Verfügung.
 - Die Information steht nicht in elektronischer Form zur Verfügung, sondern der Benutzer muß eine subjektive Beschreibung eingeben.
- *Nach dem Format der Texte:*
 - Es sind nur strukturierte Texte zugelassen.
 - Nur unstrukturierte Texte sind zugelassen.
 - Strukturierte und unstrukturierte Texte sind erlaubt, aber immer nur eine Art pro Datenbank.
 - Wie vorhin, verschiedene Textarten dürfen jedoch auch in der selben Datenbank aufscheinen.
 - Innerhalb eines Textes können oder müssen strukturierte und unstrukturierte Teile vorhanden sein.
- *Nach Art der Datenaufnahme:*
 - Die Datenaufnahme wird durch das IRS abgedeckt. Es stehen verschiedene Hilfsmittel zur Verfügung die verwendet werden müssen.
 - Die Datenaufnahme wird nicht abgedeckt. Eingabe- bzw. Editierprogramme müssen vom Benutzer selbst geschrieben werden. Das IRS umfaßt lediglich einen Lade- und einen Auswertungsteil.
 - Die Datenaufnahme kann entweder mit Hilfe des IRS erfolgen oder auch mit Hilfe einer definierten Schnittstelle umgangen werden.
- *Nach Art des Ladevorganges:*
 - Das Laden der Daten findet sofort nach jeder Änderung statt (Online-Änderung).
 - Das Laden umfaßt immer alle Texte. Die bestehende (alte) Datenbank wird vor dem Laden gelöscht.

- Kombination: Das Laden findet nicht sofort nach jeder Änderung oder Neueingabe statt. Die bestehende Datenbank wird nicht gelöscht. Es werden nur die geänderten bzw. neu aufgenommenen Texte geladen.
- *Nach der Bestimmung der Deskriptoren bei der Datenaufnahme:*
 - Die Deskriptoren müssen von Hand eingegeben werden.
 - Deskriptoren werden automatisch aus den Texten generiert, z.B. jedes Wort des Textes ergibt einen Deskriptor.
 - Deskriptoren können automatisch extrahiert werden, die dazu notwendigen Programme müssen jedoch vom Benutzer selbst geschrieben werden.
 - Deskriptoren können nur aus bestimmten Teilen des Textes extrahiert werden.
 - Deskriptoren werden gemäß einer im Voraus bestimmten Tabelle (Thesaurus) extrahiert.
 - Mischformen: Deskriptoren werden automatisch aus den Texten generiert und dem Benutzer zur Korrektur, Auswahl und Ergänzung vorgelegt.
- *Nach Art der Änderungsmöglichkeiten:*
 - Änderungen der Texte und der Deskriptoren (Art, Anzahl und Form) sind möglich.
 - Änderungen sind nur beschränkt möglich (nur Änderung der Rohdaten oder nur Änderung bestimmter Deskriptoren).
- *Nach der Art der Speicherung der Deskriptoren (Deskriptorentopologie):*
 - Flache Systeme: es besteht keine logische Verbindung zwischen den gespeicherten Deskriptoren; alle sind gleichrangig und gleichberechtigt.
 - Hierarchische Systeme: die gespeicherten Deskriptoren sind hierarchisch miteinander verknüpft.
 - Netzwerkartige Systeme: zwischen den Deskriptoren können beliebige (netzwerkartige) Strukturen und Abhängigkeiten definiert werden.
- *Nach der Verwaltung der gespeicherten Daten:*
 - Die Verwaltung erfolgt durch ein Datenbank-Management-System (DBMS).
 - Die Verwaltung wird durch das IRS selbst erledigt.
 - Die Verwaltung obliegt dem Benutzer: die Daten sind in einzelnen Dateien gespeichert, welche für den Benutzer transparent sind.

- Mischformen: Die Verwaltung wird normalerweise durch das IRS besorgt, kann aber jederzeit vom Benutzer oder von einem DBMS übernommen werden.

5.5 Beurteilung der Güte eines IRS

Die Beurteilung der Güte eines IRS und der gespeicherten Daten kann aufgrund zwei unterschiedlicher Kriterien erfolgen [Mresse 84]:

- *quantitative oder technische Gesichtspunkte*: Angaben über die Länge der Antwortzeiten des Systems oder über die Anzahl der gespeicherten Informationseinheiten.
- *qualitative Merkmale*: Diese werden im folgenden genauer beschrieben.

5.5.1 Maßzahlen

Der Benutzer eines IRS beschreibt seine Interessen durch eine Abfrage (Recherche, query). Diese Abfrage dient als Grundlage für eine Suche in der Datenbasis. Die Antwort des IRS an den Benutzer umfaßt eine Anzahl von Dokumenten oder Informationseinheiten. Die Qualität der Antwort hängt davon ab, wieviele der gespeicherten und relevanten Informationseinheiten gefunden werden und wieviele der gefundenen Informationseinheiten relevant sind. Aus der Sicht einer Abfrage kann die Gesamtmenge der gespeicherten Dokumente in 4 unterschiedliche Mengen zerteilt werden [Mresse 84]:

- *Menge Y_1* : Menge der Informationseinheiten, die bei der Recherche (zurecht) selektiert wurden und auch für die Abfrage relevant sind.
- *Menge Y_0* : Menge der Informationseinheiten, die bei der Recherche (fälschlicherweise) nicht selektiert wurden, aber für die Abfrage relevant sind.
- *Menge Z_1* : Menge der Informationseinheiten, die bei der Recherche (fälschlicherweise) selektiert wurden, aber für die Abfrage irrelevant sind.
- *Menge Z_0* : Menge der Informationseinheiten, die bei der Recherche (zurecht) nicht selektiert wurden und auch für die Abfrage irrelevant sind.

Mit diesen Mengendefinitionen lassen sich zwei Verhältniszahlen definieren [Witten 94], [Agosti 96], [Kuhlen 80]:

Präzision (Relevanzrate, precision):

Die Präzision ist die Messung der Genauigkeit der Suche (wie nahe gelangt man an das gewünschte Ziel?)

Fehler! = Fehler!

Ausbeute (Vollständigkeitsrate, recall):

Fehler! = Fehler!

Für eine perfekte Beantwortung einer beliebigen Abfrage sollten die Mengen Y_0 und Z_1 leer sein! Die Präzision und die Ausbeute konvergieren dann jeweils zur Zahl 1. Aber dieser ideale Fall tritt nur in den seltensten Fällen auf.

Bei der Bewertung der Gesamtleistung eines IRS werden ähnliche Abfragen miteinander verglichen. Unter ähnlichen Abfragen versteht man jene, die den gleichen Zweck verfolgen, aber unterschiedlich formuliert sind. Man kann für jede Antwort der ähnlichen Abfragen ein Wertepaar (Ausbeute, Präzision) berechnen und dieses graphisch mit Hilfe eines Ausbeute/Präzision-Diagrammes darstellen.

Je höher die Leistung des verwendeten IRS und je besser die Datenbasis ist, desto eher nähert man sich dem Idealpunkt mit Ausbeute=1 und Präzision=1. In der Praxis ergibt sich jedoch meist folgender Zusammenhang: Je höher die Ausbeute ist, desto geringer die Präzision und je höher die Präzision desto geringer die Ausbeute.

Eine weitere Maßzahl zur Beurteilung der Qualität eines IRS ist die Ausfallsrate (fallout ratio). Sie repräsentiert den Anteil der (fälschlich) ausgewählten irrelevanten Dokumente an allen irrelevanten Dokumenten in der Datenbasis.

Ausfallsrate (fallout ratio):

Fehler! = Fehler!

Im Idealfall hat die Ausfallsrate den Wert 0.

5.5.2 Berechnung der Maßzahlen

Die Berechnung der Präzision bietet keine Schwierigkeiten. Der Anwender kann sie während der Durchsicht der ausgewählten Dokumente vornehmen. Er braucht keine Kenntnis über die Mengen der nicht ausgewählten Dokumente (Mengen Y_0 und Z_0). Die Maßzahl der Präzision erlaubt jedoch erst im Zusammenhang mit anderen Maßzahlen Aussagen über die Qualität des IRS und der gespeicherten Daten.

Die Berechnung der Ausbeute und der Ausfallsrate setzt die Kenntnis der Mengen Y_0 und Z_0 voraus. Diese Berechnung kann auf 3 unterschiedliche Arten erfolgen:

- *Überprüfung aller gespeicherter Dokumente:* Nach jeder Abfrage werden sowohl die Dokumente der Antwort als auch alle restlichen gespeicherten Dokumente der Datenbasis auf ihre Relevanz hin untersucht. Das Resultat sind die gewünschten Mengen. Diese Methode ist sehr zeitraubend und kann daher nur für kleine Datenmengen durchgeführt werden. Allerdings ist die Berechnung der Qualitätsmasse jedoch vor allem für große und inhaltlich unbekannte Dokumentemengen interessant.

- *Anwendung von Stichprobenverfahren:* Aus den bei einer Abfrage nicht ausgewählten Dokumenten (Menge Y_0 und Z_0) werden Stichproben gezogen und auf ihre Relevanz bewertet. Durch eine anschließende Hochrechnung kann dann Y_0 und Z_0 bestimmt werden.
- *Erweiterung der Abfrage:* Nach der ersten und zu bewertenden Abfrage wird diese Abfrage so erweitert, daß mit an Sicherheit grenzender Wahrscheinlichkeit alle relevanten Dokumente gefunden werden. Dann werden alle Dokumente der beiden Antworten ausgezählt. Die Menge Y_1 entspricht den relevanten Dokumenten aus der ersten Antwort. Die Menge Y_0 ist die Differenzmenge **Fehler!**. Wobei Y_{12} die Menge der relevanten Dokumente aus der zweiten Antwort und Y_{11} die Menge der relevanten Antworten aus der ersten Antwort ist. Die Menge Z_0 kann dann auch als eine Differenzmenge berechnet werden.

Die Qualitätszahlen, die bei einem IRS mit einer Anzahl von Abfragen erreicht werden können, hängen wesentlich von folgenden 3 Punkten ab:

- *Nach welchen Kriterien werden die Deskriptoren aus den Dokumenten ausgewählt (indexing)?* Wenn generell nur Deskriptoren ausgewählt werden, die gut passen (Kerndeskriptoren), dann wird die Präzision verbessert. Werden dagegen auch Deskriptoren aufgenommen, die nur teilweise zutreffen (Randdeskriptoren), dann wird die Ausbeute erhöht.
- *Wie wird die Abfrage formuliert?* Es ist leicht eine Abfrage so zu formulieren, daß die Ausbeute gleich 1 wird, wenn dafür dann eine sehr geringe Präzision in Kauf genommen wird.
- *Welche Möglichkeiten bietet das IRS?* Schließlich hat auch das IRS selbst Möglichkeiten, die es bei der Aufnahme und bei der Formulierung der Abfrage bietet, einen Einfluß auf die erreichbaren Verhältniszahlen zu nehmen. Mittel zur Fehlererkennung bei manueller Eingabe, zur Erkennung ähnlicher Deskriptoren bei der Aufnahme oder bei der Abfrage, usw. erhöhen die Ausbeute bei gleichbleibender oder verbesserter Präzision.

Die Berechnung aller Maßzahlen obliegt dem Benutzer, er muß schließlich eine subjektive Bewertung über die Relevanz der einzelnen Dokumente durchführen. Die größte Aussagekraft haben Maßzahlen bei Volltext-IRS, die zusätzlich einen Mechanismus über eine automatische Deskriptorauswahl verfügen. Die Menge der gespeicherten Dokumente muß sehr groß sein, um statistisch signifikante Aussagen tätigen zu können.

5.6 Deskriptorenauswahl

Unter dem Begriff Deskriptorenauswahl (indexing) werden alle Methoden und Verfahren zusammengefaßt die der Bestimmung von Deskriptoren zur Charakterisierung eines Dokumentes dienen [Mresse 84].

5.6.1 Manuelle Auswahl

Bei der manuellen Auswahl (manual indexing) werden die relevanten Merkmale eines Dokumentes vom Benutzer ausgewählt. Es besteht aber auch die Möglichkeit diese manuelle Auswahl mit einer automatischen Auswahl zu kombinieren: Das IRS schlägt eine Liste von Deskriptoren vor und der Benutzer kann Elemente in diese Liste hinzufügen oder auch streichen.

Vorteile: Es werden auch Terme berücksichtigt, die nicht direkt aus dem Dokument hervorgehen, z.B. Deskriptoren die Urteile über die Qualität des Dokumentes oder Hinweise auf die Verwendung des Dokumentes geben. Diese Art der Auswahl ist auch sehr einfach in das IRS zu implementieren. Ihre Anwendung findet sich in Bereichen der Datenerfassung an denen nur wenige und gut ausgebildete Benutzer arbeiten.

Nachteile: Die Datenaufnahme erfordert einen Mehraufwand durch die zweimalige Eingabe der relevanten Terme (als Text und als Deskriptor).

5.6.2 Automatische Auswahl

Hier entfällt das Eingeben der Deskriptoren. Diese werden automatisch (automatic indexing) vom IRS nach einem bestimmten Auswahlverfahren gewählt. Alle automatischen Auswahlverfahren können erst dann effizient eingesetzt werden, wenn sämtliche Deskriptoren automatisch auf ihre Grundform oder ihren Wortstamm reduziert werden können. Unterschiedliche Schreibweisen sind auf eine Standardschreibweise zu reduzieren. Einige übliche Methoden zur automatischen Deskriptorenauswahl:

- *Auswahl aufgrund von Wortlisten:* Nur diejenigen Wörter werden als Deskriptoren verwendet, die auch in einer Wortliste (Thesaurus) enthalten sind. Der Thesaurus muß unabhängig von der Datenbank im Voraus definiert werden.
- *Auswahl mit Hilfe von Stoppwortlisten:* Aus den Texten werden alle Wörter ausgewählt, die nicht in einer Stoppwortliste (auch Anti-Thesaurus) vorkommen. Der Vorteil dieser Methode liegt darin begründet, daß durch die Stoppwortliste Wörter wie (und, der, die, das...) nicht als Deskriptoren aufgenommen werden. Nachteil: Eine Suche nach z.B. dem Hamlet-Zitat "Sein oder Nichtsein" funktioniert dadurch nicht [Frakes 92].

- *Auswahl aufgrund einer weitergehenden Textanalyse:* Als Ergänzung zu einer der genannten automatischen Verfahren kann der gesamte Text zur Bildung von zusammengesetzten Deskriptoren überprüft werden. Durch eine Analyse des Dokumentes kann ein einzelnes Wort, das keinen Wert als Deskriptor hat, zu einem wichtigen Deskriptor ergänzt werden. Der Effekt dieser Vorgehensweise ist, daß ein zu allgemeiner Deskriptor durch einen spezielleren ersetzt wird. Die Auswahl der Dokumente wird dadurch präziser, die Ausbeute sinkt aber gleichzeitig.
- *Gewichtete Auswahl:* Bei der gewichteten Auswahl wird versucht, den einzelnen Deskriptor im Kontext seiner Umgebung zu betrachten. Als Umgebung wird entweder das Dokument, die Abfrage oder die gesamte Datenbank gesehen. Diese Gewichtung der Deskriptoren setzt, soweit sie bei der Datenaufnahme oder beim Laden der Daten in die Datenbank vorgenommen wird, ein geeignetes Verfahren zur Selektion der möglichen Deskriptoren aus den Texten voraus. Nach dieser Erstausswahl kann anhand einer Gewichtung eine definitive Auswahl und Speicherung vorgenommen werden. Die Gewichtung bei der Abfrage setzt ebenfalls ein geeignetes Verfahren zur Deskriptorauswahl voraus, dieses hat aber keinen Einfluß auf die gespeicherten Deskriptoren.

5.6.3 Probleme bei der Deskriptorauswahl

Einige der Probleme die in einem IRS auftreten könnten [Lancaster 78]:

- Die Dokumente sind in mehreren verschiedenen Sprachen gespeichert, aber pro Dokument nur in einer Sprache. Das bedeutet, Dokumente werden abhängig von der Sprache, in der die Abfrage formuliert wurde, gefunden. Eine mögliche Abhilfe wäre die automatische Übersetzung der Abfrage in mehrere Sprachen.
- Das gesamte IRS besteht aus mehreren einzelnen Datenbanken, die unterschiedliche Abfragemethoden bereitstellen. Eine Lösung für dieses Problem wäre eine automatische Konvertierung der Abfrage in die jeweils notwendige Syntax.
- Der Benutzer wird mit zu vielen Dokumenten als Ergebnis einer Abfrage überhäuft. Er muß sich sehr viele Dokumente betrachten, bevor er das für ihn relevante Dokument findet. Dies entsteht entweder durch eine zu vage formulierte Abfrage, oder durch eine zu große Datenbank über ein Themengebiet, wodurch gleiche Deskriptoren in zu vielen Dokumenten auftreten, und dadurch eine einfache Unterscheidung bzw. Selektion verhindern.

5.6.4 Speicherung der Deskriptoren (Methode des inverted index)

In der Geschichte der Erforschung von IRS wurde immer viel Aufwand getätigt, um die Qualität der zurückgelieferten Dokumente zu verbessern [Agosti 96]. Der Aufwand um

die Suche effektiv auszuführen wurde für lange Zeit vernachlässigt. Die Zeit, um eine sequentielle Suche durch alle Dokumente nach einem bestimmten Wort durchzuführen, ist viel zu groß, um in Echtzeit mit diesem System arbeiten zu können. Um dieses Problem zu umgehen, wurde die Methode des inverted index entwickelt: Es wird jeder Deskriptor mit Verweis auf alle Dokumente in denen sich dieser Deskriptor befindet gespeichert. Bei jedem Laden eines Dokumentes oder bei jeder Änderung wird der Inhalt dieses inverted index aktualisiert. Dies hat den Vorteil, daß die Suche nach allen Dokumenten, die einen bestimmten Deskriptor enthalten, sehr rasch möglich ist. Weiters ist auch eine boolesche Kombination von Deskriptoren einfach zu implementieren. Im Vergleich zu einer sequentiellen Suche ist die Methode des inverted index wesentlich schneller, hat aber den Nachteil, daß beim Laden eines Dokumentes in das IRS das Erstellen des inverted index eine gewisse Rechenzeit in Anspruch nimmt.

5.6.5 Maßzahlen zur Gewichtung von Deskriptoren

Diese Maßzahlen unterscheiden sich in der Aussage, die sie ermöglichen, im Aufwand, der für ihre Berechnung notwendig ist und im Zeitpunkt ihrer Verwendung. Alle Maßzahlen sind Relevanzfaktoren, d.h. sie versuchen etwas über die Wichtigkeit eines Deskriptors auszusagen.

dokumentweise Betrachtung:

Bei der dokumentweisen Betrachtung wird für jeden Deskriptor ein Gewicht berechnet, welches eine Aussage über die Wichtigkeit des Deskriptors innerhalb des Dokumentes trifft. Dieses Gewicht wird aufgrund der Häufigkeit des Erscheinens des Deskriptors im Dokument berechnet. Das Gewicht eines Deskriptors D_i im Dokument T_j wird definiert als:

$$G =$$

Fehler!

Das Gewicht G kann Werte zwischen 0 und 1 annehmen. Je näher das Gewicht bei 1 liegt, desto wichtiger ist der Deskriptor in diesem Dokument. Bei der Deskriptorenauswahl werden alle Terme, deren Gewicht einen bestimmten Grenzwert nicht überschreiten, ausgeschieden. Die Berechnung des Gewichtes ist nur in Volltext-IRS sinnvoll.

abfrageweise Betrachtung:

Hier werden anstelle eines einzelnen Dokumentes alle Dokumente einer Abfrage berücksichtigt.

Definition des Relevanzfaktors:

RF =

Fehler!

Der Relevanzfaktor ist bei der Modifikation und Wiederholung von Abfragen von Bedeutung. Um die Präzision einer Abfrage bei gleichbleibender Ausbeute zu erhöhen, werden in der modifizierten Abfrage nur Deskriptoren verwendet, deren Relevanzfaktor über einen bestimmten Grenzwert liegt. Die Verwendung dieses Faktors kann bei großen Datenmengen auch in einem Nicht-Volltext-IRS sinnvoll sein.

dokumentspezifische Betrachtung:

Diese Maßzahl entstand aus der Erkenntnis, daß die Wichtigkeit der Deskriptoren eines Dokumentes vom Inhalt anderer Dokumente abhängig ist, z.B. ein Deskriptor "computer" in einer Artikelsammlung über Datenverarbeitung hat einen geringen Wert, der gleiche Deskriptor in einer Artikelsammlung über Medizin einen wesentlich größeren Wert.

Per definitionem wird die Inverse Document Frequency eingeführt:

IDF =

Fehler!

Die IDF eines Deskriptors D_i im Dokument T_j steigt, je häufiger er in diesem Dokument erscheint. Sie sinkt, wenn er aber in vielen Dokumenten vorkommt. Je kleiner der Wert von IDF ist, desto unwichtiger ist dieser Deskriptor. Die IDF sollte im Idealfall für alle in der Abfrage verwendeten Deskriptoren in allen ausgewählten Dokumenten einen möglichst hohen Wert erreichen. Es kann wieder ein Grenzwert festgelegt werden und alle Deskriptoren unter diesem Grenzwert werden ausgeschlossen. Die IDF ist nur in Volltext-IRS mit automatischer Indexierung sinnvoll.

abfragespezifische Betrachtung:

Bei der Gewichtung von ganzen Dokumenten wird mit folgender Maßzahl gearbeitet. Sie wird bei jeder Abfrage für alle Deskriptoren, die in Verbindung mit einem der selektierten Dokumente stehen, berechnet. Die Berechnung von L setzt die Bewertung aller selektierten Dokumente (oder einer Stichprobe aus der Menge der selektierten Dokumente) durch den Benutzer voraus.

Die Definition von L lautet:

L_{iq} =

Fehler!

Je größer L für einen bestimmten Deskriptor ist, desto wichtiger ist er. Die Berechnung von L wird während der Abfrage vorgenommen. Sie ist auch in Nicht-Volltext-IRS sinnvoll, wenn die Menge der gespeicherten Dokumente groß genug ist.

Zusammenfassung der erläuterten Maßzahlen:

Maßzahl	Wert für wichtigen Deskriptor	Wert für unwichtigen Deskriptor	Berechnung	Aussage
G	gegen 1	gegen 0	beim Laden	ein wichtiger Deskriptor tritt nur in einem Dokument auf
RF	gegen 1	gegen 0	bei der Abfrage	ein wichtiger Deskriptor ist eng mit der Abfrage verbunden
IDF	gegen unendlich	gegen 1	beim Laden	ein wichtiger Deskriptor ist eng mit dem Dokument verbunden
L	gegen unendlich	gegen 0	bei der Abfrage	ein wichtiger Deskriptor bezeichnet den Sinn der Abfrage sehr genau

5.6.6 Zusammenfassung

Ein mögliches Vorgehen in der Praxis bei der Deskriptorenauswahl:

1. Die einzelnen Wörter im Volltext identifizieren.
2. Mit einer Stoppwortliste die nicht relevanten Wörter ausscheiden [Agosti 96].
3. Automatische Wortstammextraktion, abschneiden von Prä- und Suffixen (stemming) [Agosti 96], [Salton 68], [Frakes 92].
4. Zu seltende Deskriptoren werden unter Anwendung des Thesaurus durch andere Deskriptoren ersetzt.
5. zu häufige Deskriptoren werden mit anderen Wörtern des gleichen Satzes kombiniert.
6. Berechnung der IDF-Funktion für alle Wortstämme und alle abgeleiteten und zusammengesetzten Deskriptoren für alle Texte.

Einige dieser Methoden sind sehr zeitintensiv, wie z.B. die textbezogene, wiederholte Neuberechnung der IDF, die bei jeder Neuaufnahme für alle Dokumente durchgeführt werden muß.

Andere erfordern wiederum viel Speicherplatz: Der Umfang von Thesauri kann sehr groß werden, und die Wortstammsuche erfordert unter Umständen das Vorhandensein ganzer Wörterbücher.

5.7 Auswahl mit Rückkopplung (relevance feedback)

Unter dem Namen Auswahl mit Rückkopplung werden alle Methoden zusammengefaßt, die nach folgendem Schema arbeiten [Mresse 84], [Agosti 96], [Salton 68]:

1. Dem Benutzer wird aus der Menge der gefundenen Dokumente eine zufällige Auswahl zur Begutachtung vorgelegt.
2. Der Benutzer beurteilt die vorgelegten Dokumente, indem er sie als relevant oder nichtrelevant bezeichnet.
3. Das IRS versucht aufgrund der Auswahl, auch alle anderen gefundenen Dokumente zu klassifizieren und dem Benutzer in entsprechender Reihenfolge zu präsentieren.
4. Wenn die Antwort noch unzureichend ist, kann das IRS versuchen, aufgrund der Auswahl eine neue Abfrage zu formulieren.

5.7.1 Gewichtung der selektierten Dokumente

Aufgrund der Benutzerauswahl wird für jeden Deskriptor D_i , der im Zusammenhang mit einem Dokument in der Abfrage A_q steht, ein Relevanzfaktor L berechnet. Hier ist aber zu beachten, daß sich die Anzahl relevanter und nicht relevanter Dokumente in der Berechnungsformel für L_{iq} auf die bewertete Stichprobe aus der Abfrage beziehen, d.h. diese Zahlen sind nicht identisch mit den vorhin erwähnten Mengen Y und Z . Aufgrund der Maßzahl L werden in einem ersten Schritt ein dokumentspezifisches Gewicht für jeden Deskriptor berechnet. Es wird L_i mit der Häufigkeit des Deskriptors in jedem Dokument multipliziert. Anschließend wird der Deskriptor-Relevanz-Faktor (DRF) wie folgt berechnet:

$$DRF_{ij} = L_i * \text{Häufigkeit von } D_i \text{ in } T_j$$

Für jedes Dokument T_j berechnet man als Gewichtung WT_j die Summe über alle DRF_{ij} :

$$WT_j = \sum DRF_{ij}$$

Bei diesen Berechnungen sollte man auf zwei Spezialfälle Rücksicht nehmen:

- Wenn ein Deskriptor nur in relevanten Dokumenten erscheint, geht L_i gegen unendlich aufgrund einer Division durch 0.
- Wenn alle Dokumente der Auswahl als relevant bezeichnet werden, ist in weiterer Folge keine sinnvolle Berechnung möglich.

Die berechnete Maßzahl WT kann dazu verwendet werden, dem Benutzer die selektierten Dokumente ihrer Wichtigkeit nach zu präsentieren. Der Hauptzweck der Berechnung von WT liegt jedoch in einer entsprechenden Modifikation der Abfrage. Bei einer Wiederholung der Abfrage soll die Ausbeute bei gleichbleibender oder erhöhter Präzision verbessert werden. Dazu werden alle Dokumente in der Datenbank gesucht, die mit mindestens einem der relevantesten Deskriptoren in Verbindung stehen. Für sämtliche Deskriptoren dieser neuen Dokumente wird wiederum die DRF berechnet und für alle Dokumente zur Gewichtung WT aufsummiert. Es entsteht dadurch eine neue Folge von Dokumenten, die dem Benutzer nach absteigender Relevanz sortiert präsentiert werden. Dieser Vorgang kann mehrmals wiederholt werden.

5.8 Verwendungsarten von Thesauri

Unter einem Thesaurus versteht man jede Art von Wörterbuch. Thesauri werden im Zusammenhang mit IRS auf verschiedene Arten verwendet [Mresse 84], [Salton 68], [Frakes 92]:

- als vordefinierte Deskriptorenliste
- als Synonymwörterbuch
- zur Gruppierung, Klassifizierung oder Strukturierung von Deskriptoren

5.8.1 vordefinierte Listen

Das Hauptproblem bei Thesauri liegt in der Notwendigkeit, den Thesaurus im Voraus zu definieren. In verschiedenen Fachgebieten werden bereits umfassende Thesauri angeboten. Der Vorteil dieser vordefinierten Listen liegt darin, daß bei der Datenaufnahme die terminologische Kontrolle, d.h. die Kontrolle über die einheitliche Verwendung der Terme generell verbessert werden kann.

5.8.2 Synonymwörterbuch

Als Synonyme werden unterschiedliche Wörter mit gleicher Bedeutung bezeichnet. Der Thesaurus hat bei der Verwendung als Synonymwörterbuch die Aufgabe synonyme Wörter zu gruppieren. Alle Synonyme bilden eine Klasse. Es werden drei Varianten bei dieser Art der Verwendung eines Thesaurus unterschieden:

- *Variante 1:* Jeder Deskriptor wird bereits bei der Datenaufnahme durch die anderen Deskriptoren seiner Klasse erweitert. Ein Dokument wird somit auch durch die Eingabe des synonymen Termes gefunden. Generell wird dadurch die Ausbeute erhöht. Der Thesaurus wird nur bei der Datenaufnahme verwendet.
- *Variante 2:* Um die explizite Speicherung aller Klassenmitglieder eines Deskriptors zu vermeiden, kann der Thesaurus auch nur bei der Abfrage fakultativ als Filter verwendet werden. Der große Vorteil dieser Variante liegt in der beibehaltenen Flexibilität. Es können weiterhin Deskriptoren nach Belieben "erfunden" werden. Die Menge der Deskriptoren wird nicht durch Synonyme vergrößert. Die Abfrage wird nicht durch eine zwangsweise Verwendung eines Thesaurus verändert und verlangsamt. Der Nachteil ist, daß die Kontrollmöglichkeiten über fehlerhafte Deskriptorenangaben bei der Datenaufnahme begrenzt sind weil sie erst bei der Abfrage mit den Einträgen im Thesaurus verglichen werden.
- *Variante 3:* Wenn der Thesaurus benutzt wird, um anstelle des aufgenommenen Deskriptors eine Art Stammwort aus der Deskriptorenklasse zu speichern, spricht man von einer Speicherung in kanonischer Form. Da die aus dem Deskriptor generierten kanonischen Formen dem Benutzer nicht unbedingt bekannt sind, muß auch bei der Abfrage jeder Deskriptor durch den selben Thesaurus gefiltert werden. Änderungen im Thesaurus können schwerwiegende Folgen haben: Sie können zu falschen Interpretationen oder zur Unauffindbarkeit führen. Die Vorteile sind, daß die Menge der gespeicherten Deskriptoren kleiner wird, da pro Klasse nur ein Deskriptor gespeichert wird. Bei der Abfrage nach Stammwörtern kann der Thesaurus umgangen werden. Es ist dann mit optimalen Antwortzeiten zu rechnen. Als Nachteile treten in Erscheinung, daß die ursprüngliche Form der Eingabe verloren geht, daß das Konzept sehr starr ist und bei der Aufnahme und bei der Abfrage muß derselbe Thesaurus verwendet werden.

5.9 Erkennung physisch ähnlicher Zeichenketten

Um die Qualität der Antwort eines IRS zu verbessern besteht oft der Wunsch nach nicht exakten Vergleichsmethoden zur Erkennung ähnlicher Zeichenketten. Es soll die Möglichkeit bestehen, Abfragen nur vage zu formulieren bzw. soll mehr als nur die genaue Eingabe berücksichtigt werden. Die physische und semantische Umgebung eines Deskriptors soll miteinbezogen werden. Ähnlichkeiten zwischen Deskriptoren können auf unterschiedlicher Stufe behandelt werden [Mresse 84]:

physische Ähnlichkeiten (similarity):

- Angabe von Metazeichen
- Vergleich ähnlicher Zeichenketten

semantische Ähnlichkeiten (equivalence) auf der Basis einzelner Deskriptoren:

- Deskriptorauswahl
- Thesaurus bei der Datenaufnahme
- Thesaurus bei der Abfrage

semantische Ähnlichkeiten auf der Basis ganzer Texte bzw. ihrer Deskriptoren:

- Deskriptorengewichtung
- Rückkopplungstechniken

Im folgenden wird die Möglichkeit einer unterschiedlichen Eingabe gleicher Deskriptoren genauer untersucht. Es läßt sich eine Unterteilung in 2 Punkte vornehmen:

unterschiedliche Eingaben aufgrund sprachlicher Gegebenheiten:

- das gleiche Wort wird in verschiedenen Formen eingegeben (Prä- und Suffixproblem)
- das phonetisch gleiche Wort wird unterschiedlich geschrieben
- eine Rechtschreibreform ändert per Definitionem die Schreibweise von Wörtern

unterschiedliche Eingaben als Folge von Fehlern:

- auslassen eines Buchstabens
- einfügen eines falschen Buchstabens
- vertauschen von zwei am Eingabegerät (Tastatur) benachbarter Buchstaben

In der Literatur werden etliche Lösungen dieses similarity-Problemes vorgeschlagen. Doch sind die meisten sehr aufwendig, da der zugrundeliegende Algorithmus meist die Ordnung $O(n^2)$ hat.

5.9.1 Unterschiedliche Eingaben aufgrund sprachlicher Gegebenheiten

Bei unterschiedlichen Eingaben aufgrund sprachlicher Gegebenheiten sind die Eingaben an sich korrekt, erschweren jedoch die Abfrage durch ihre äußere Erscheinungsform.

Angabe von Metazeichen:

Bei dieser Methode werden einzelne Buchstaben der Abfrage durch spezielle Zeichen (z.B. ? oder *) ersetzt. Diese Metazeichen werden bei internen Vergleichen als eine Art Joker verwendet: Sie können stellvertretend für eine bestimmte oder beliebige Anzahl von anderen Zeichen stehen. In eckigen Klammern können auch mehrere Buchstaben zur Auswahl angeboten werden. Ein Beispiel: In einer Datenbank sollen alle Artikel im Zusammenhang mit Organisation und Datenbank gefunden werden. Die Abfrage lautet:

„da*ba* && organi[sz]lation“. Diese Form der Abfrage berücksichtigt sowohl die deutschen Deskriptoren "Organisation" und "Datenbank" als auch die englischen "organization" und "database".

Soundex-Methode:

Dies ist eine der ersten Methoden zur Ermittlung von Ähnlichkeiten. Sie reduziert alle Deskriptoren auf Zeichenketten mit einem alphanumerischen und maximal 3 numerischen Zeichen. Auf diese Weise werden ähnlich klingende und teilweise falsch geschriebenen Wörter auf gleiche Formen zurückgeführt. Der erste Buchstabe des Deskriptors wird als erste Stelle für den Soundex-Code verwendet. Für die Transformation aller weiteren Buchstaben gilt (es ist zu beachten, daß sich diese Tabelle bei unterschiedlichen Sprachen ändern kann):

Die Buchstaben:	ergeben:
a e i o u h w y	0
b f p v	1
c g j k q s x z	2
d t	3
l	4
m n	5
r	6

Bei der Transformation werden alle Nullen ausgeschieden, hintereinander vorkommende gleiche Zahlen auf eine Zahl reduziert und der ganze Code auf maximal 4 Zeichen beschränkt. Beispiel: Die beiden Namen Neumann und Newman werden als gleich erkannt:

Ausgangspunkt: Neumann - Newman

Transformation: N005055 - N00505

Nullen entfernen: N555 - N55

gleiche Zahlen entfernen: N5 - N5

→ Neumann und Newman wurden als gleich erkannt

5.10 Vergleich von Zeichenketten

Eine der häufigsten Operationen in IRS ist der Vergleich von Zeichenketten. Es ist daher notwendig die verfügbaren Algorithmen für dieses Problem genauer zu analysieren. Das Problem in allgemeiner Form lautet [Mresse 84], [Frakes 92]:

Finde in einer Zeichenkette eine andere, kürzere Zeichenkette.

5.10.1 Straight Forward (SF) - Algorithmus

Nach der SF-Methode wird jeweils das erste Zeichen der gesuchten Zeichenkette im zu durchsuchenden Text gesucht. Wird eines gefunden, dann wird ein weiteres Zeichen verglichen. Die Suche ist dann erfolgreich, wenn alle gesuchten Zeichen mit denen im Text übereinstimmen. Der SF-Algorithmus zeigt im schlechtesten Fall (worst case) ein Verhalten von $O(\text{Länge der gesuchten Zeichenkette} * \text{Länge des gesamten Textes})$ wenn zwei Bedingungen eintreffen:

1. Wenn die gesuchte Zeichenkette irgendwo im Text und nicht nur am Anfang gesucht werden muß.
2. Wenn der Vergleich immer erst beim letzten zu vergleichenden Zeichen abgebrochen werden kann (Bsp.: suche die Zeichenkette „aa.ab“ im Text „aaaaaaaa.....aaaaaaaa“).

Da diese zweite Bedingung selten erfüllt ist, wird der Algorithmus in der Regel ein günstigeres Verhalten zeigen. Wenn auch die erste Bedingung nicht erfüllt ist, kann die Suche in der Regel schon nach wenigen Zeichen abgebrochen werden.

5.10.2 KMP-Algorithmus

Dieser Algorithmus geht auf Knuth, Morris und Pratt zurück. Er setzt sich das Ziel, mehrfache Vergleiche gleicher Positionen zu vermeiden und die gesuchte Zeichenkette nach jedem mißlungenen Versuch um mehr als nur eine Position im zu durchsuchenden Text weiterzurücken.

In einer Tabelle wird für jede Position der gesuchten Zeichenkette festgehalten auf welche Position sie bei einer Nichtübereinstimmung verschoben werden darf. Im schlechtesten Fall zeigt dieser Algorithmus ein Verhalten von $O(\text{Länge der gesuchten Zeichenkette} + \text{Länge des gesamten Textes})$ was bedeutend besser ist als der vorherige SF-Algorithmus.

5.10.3 BM-Algorithmus oder Algorithmus von Boyer und Moore

Im Unterschied zum KMP-Algorithmus beginnt die Suche im BM-Algorithmus jeweils beim letzten Zeichen der zu suchenden Zeichenkette, die Suche geht immer rückwärts. Zur Positions- und Sprungkontrolle werden mehrere Tabellen verwendet.

Dieser Algorithmus benötigt weniger als $O(\text{Länge der gesuchten Zeichenkette} + \text{Länge des gesamten Textes})$ Vergleiche.

5.10.4 AC-Algorithmus

Der Algorithmus von Aho und Corasick beruht auf ähnlichen Überlegungen wie sie im KMP-Algorithmus angestellt wurden: Jedes Zeichen eines Textes soll nur einmal verglichen werden. Im Fall des Nichtübereinstimmens wird eine Sprungtabelle (failure-function) konsultiert, aus welcher die weitere Vorgangsweise ersichtlich ist.

Zusätzlich verfolgt der AC-Algorithmus das Ziel, mehrere Zeichenketten gleichzeitig in einem Text zu suchen. Zu diesem Zweck wird eine zweite Sprungtabelle aufgebaut (goto-function), in welcher die Zusammenhänge zwischen den gesuchten Teilzeichenketten festgehalten werden. Eine dritte Tabelle (output-function) enthält alle gesuchten Teilzeichenketten. Die Anwendung dieses Verfahrens lohnt sich nur, wenn die gesuchten Teilzeichenketten gemeinsame Unterzeichenketten beinhalten.

5.11 abschließende Bemerkungen

Eines der ungewöhnlichsten Dinge im Forschungsbereich des Information Retrievals ist es, daß der Technologietransfer von der Forschung zu einem fertigen Endprodukt sehr langsam ist [Agosti 96]. Sehr viel langsamer als z.B. im verwandten Bereich der Datenbankverwaltung. Es in den letzten Jahren begannen die großen Informationssysteme die Antworten auf eine Abfrage nach Relevanz sortiert auszugeben. Dieses Verhalten könnte sich aber in Zukunft ändern, da die Menge an angebotener Information, die über Computer abfragbar ist, sehr stark anwächst und es daher notwendig ist, effiziente IRS zur Verfügung zu stellen.

6 QUERYCOLLECTION FÜR HYPERWAVE

6.1 Motivation

In einer speziellen Collection, genannt Querycollection soll die Möglichkeit bestehen, beliebige Objekte (Dokumente verschiedensten Types) mit genau spezifizierter Metainformation zu speichern. Dabei soll die Struktur der Metainformation für alle Objekte dieser Collection gleich sein.

Ein Anwendungsbeispiel wäre eine Bibliotheksdatenbank, die folgendermaßen aufgebaut sein könnte:

Für jedes Buch wird eine kurze Zusammenfassung (Abstract) erzeugt. Beim Speichern jeder dieser Zusammenfassungen werden noch zusätzliche Informationen wie z.B. Autor, Erscheinungsjahr, Verlag und Auflage vom System abgefragt, ev. einer Plausibilitätskontrolle unterzogen und dann gemeinsam mit der Zusammenfassung als ein Textdokument mit zusätzlicher Metainformation in der Querycollection gespeichert.

D.h. in der Querycollection befinden sich die Zusammenfassungen aller Bücher und deren Metainformation.

Ein Bibliotheksbenutzer der ein bestimmtes Buch sucht hat folgende Möglichkeiten durch Betreten der Querycollection:

- Er führt eine Volltextsuche in dieser Querycollection durch und je nachdem, wie gut die Zusammenfassungen sind, werden ihm entsprechende Ergebnisse angeboten.
- Er führt eine Suche über die Metainformation (Autor, Erscheinungsjahr, Verlag,..) durch.

6.2 Vergleich Querycollection mit einer normalen Collection in Hyperwave

normale Collection

1. Die Metainformation der Dokumente in der Collection kann beliebig sein.
2. Die Metainformation muß nicht angegeben werden.
3. Beim Betreten der Collection werden alle Dokumente automatisch aufgelistet.
4. Es können alle unter Hyperwave definierten Dokumenttypen verwendet werden.

Querycollection

1. Die Metainformation der Dokumente in der Querycollection ist genau festgelegt.
2. Die Metainformation muß beim Einfügen eines Dokumentes immer angegeben werden.
3. Beim Betreten der Querycollection wird ein Suchformular angezeigt, mit dem nach bestimmten Dokumenten in der Querycollection gesucht werden kann.
4. Es können alle unter Hyperwave definierten Dokumenttypen verwendet werden.

6.3 Typischer Verlauf der Erstellung und Benutzung einer Querycollection

6.3.1 Erstellung

- Der Autor überlegt sich welche Metainformationen zu den Dokumenten benötigt werden.
- Der Autor erzeugt eine Querycollection am Hyperwave-Server.
- Mit Hilfe eines Queryform-Wizards erstellt er das Formular für die Eingabe und die Suche der Metainformation.
- Dieses Formular wird als HTML-Dokument mit dem Attribut „PresentationHint=FullCollectionHead“ in der Querycollection gespeichert.
- Bei jedem Dokument, daß der Autor in die Querycollection einfügt, muß er zusätzlich das Formular für die Metainformation ausfüllen. Danach wird das Dokument mit der Metainformation die sich im Attribut Keywords befindet am Server gespeichert.

6.3.2 Benutzung

- Beim Betreten der Querycollection erscheint nur das Suchformular für die Metainformation.
- Durch Eingabe von Suchkriterien und Starten der Suche werden jene Dokumente in der Querycollection angezeigt, deren Metainformation den Suchkriterien entspricht.

6.4 Anwendungsbeispiele

6.4.1 Bibliotheksdatenbank

Das Dokument selbst ist eine textuelle Zusammenfassung des Inhaltes des Buches.

Möglichkeiten für die Metainformation wären:

Informationen über das Buch: Titel, Autoren, Verlag, Erscheinungsjahr, Auflage, Kaufpreis.

Informationen über die Verwaltung des Buches: Signatur, Fach- oder Themenbereich, Datum des Ankaufes, Name und Adresse des Ausleihers des Buches, Datum der Ausleihe.

6.4.2 Personendatenbank

Das Dokument selbst ist ein Bild der Person.

Metainformation:

Vorname, Zuname, Wohnadresse, Geburtsdatum, Geschlecht, Arbeitsbereich, Qualifikationen, ...

6.4.3 Audio/Videodatenbank

Das Dokument selbst ist eine Audio- oder Videodatei.

Metainformation:

Themenbereich, Qualität der Aufnahme, Format, Aufnahmedatum, zusätzliche Beschreibung der Datei, Name der Person die die Aufnahme durchgeführt hat, ...

6.5 Queryform-Wizard

6.5.1 Was ist ein Wizard?

Ein Wizard ist ein Programm, daß den Benutzer bei der Bewältigung komplexer Aufgaben unterstützt. Üblicherweise besteht ein Wizard aus mehreren Bildschirmseiten, die in sequentieller Reihenfolge betrachtet werden können. Auf diesen einzelnen Seiten werden Informationen vom Benutzer erfragt, die zur Lösung der Aufgabe benötigt werden. Die Abfrage dieser benötigten Informationen erfolgt in einer möglichst einfachen Weise, sodaß auch Benutzer mit wenig Hintergrundwissen über die zu lösende Aufgabe, diese einfach und effizient bewältigen können.

6.5.2 Ziel des Queryform-Wizards

Dieser Wizard soll die Möglichkeit bieten, sehr einfach ohne Kenntnisse über Programmierung und Gestaltung von WWW-Seiten ([Chandler 95] [Wilson 95] [Mary 96] [Niederst 96]) mit HTML (Hyper Text Markup Language [Jones 95] [Jorge 96] [Graham 95]) ein ansprechendes Such- und Eingabeformular für die Metainformation zu gestalten.

6.5.3 Funktionsweise

Ablauf der Erstellung eines Such- bzw. Eingabeformulares mit Hilfe des Queryform-Wizards:

Durch Erzeugen einer Querycollection am Hyperwave-Server wird automatisch der Queryform-Wizard gestartet.

Die einzelnen Seiten dieses Wizards sind in 3 Bereiche gegliedert:

- Im unteren Bereich befinden sich die Navigationselemente um sich durch die einzelnen Seiten des Wizards zu bewegen („Back“, „Next“), den Wizard abubrechen („Cancel“), die Hilfe aufzurufen („Help“), oder den Queryform-Wizard sofort das Formular mit den aktuellen Einstellungen fertigstellen lassen („Finish“)
- Im rechten Fenster befindet sich ein HTML-Viewer, in dem nach jedem Betätigen des Kontrollelementes „Refresh“ das aktuelle Aussehen des Such- bzw. Eingabeformulares dargestellt wird. Mit dem Kontrollelement „Maximize“ kann das Fenster des HTML-Viewers vergrößert werden, um einen besseren Überblick zu erhalten.
- Im linken Bereich des Wizards befinden sich die Eingabeelemente zur Gestaltung des Formulars.

6.5.3.1 Einstiegsseite

Auf der ersten Seite des Wizards, erhält der Benutzer eine kurze Erklärung über die Nutzungsmöglichkeiten dieses Wizards und wird dann gefragt, ob er ein neues Formular erzeugen möchte, oder ein bereits existierendes Formular verändern. Wenn ein bereits existierendes Formular verändert werden soll, so erscheint eine Dialogbox mit der man das gewünschte Formular auswählen und laden kann. Durch Drücken des Kontrollelementes „Next“ gelangt man zur nächsten Seite des Wizards.

Einschränkungen:

Wenn man sich für eine der beiden Möglichkeiten entschieden hat, kann man diese Entscheidung nicht mehr rückgängig machen. Man muß den „Cancel“-Button drücken, und den Queryform-Wizard erneut starten.

Alle weiteren Entscheidungen bezüglich Inhalt und Design des zu erstellenden Formulars sind reversibel. Die Kontrollelemente „Refresh“ und „Maximize“ sind auf dieser Bildschirmseite noch nicht aktiviert, da noch keine Entscheidung über das Aussehen des Formulars getroffen wurde.

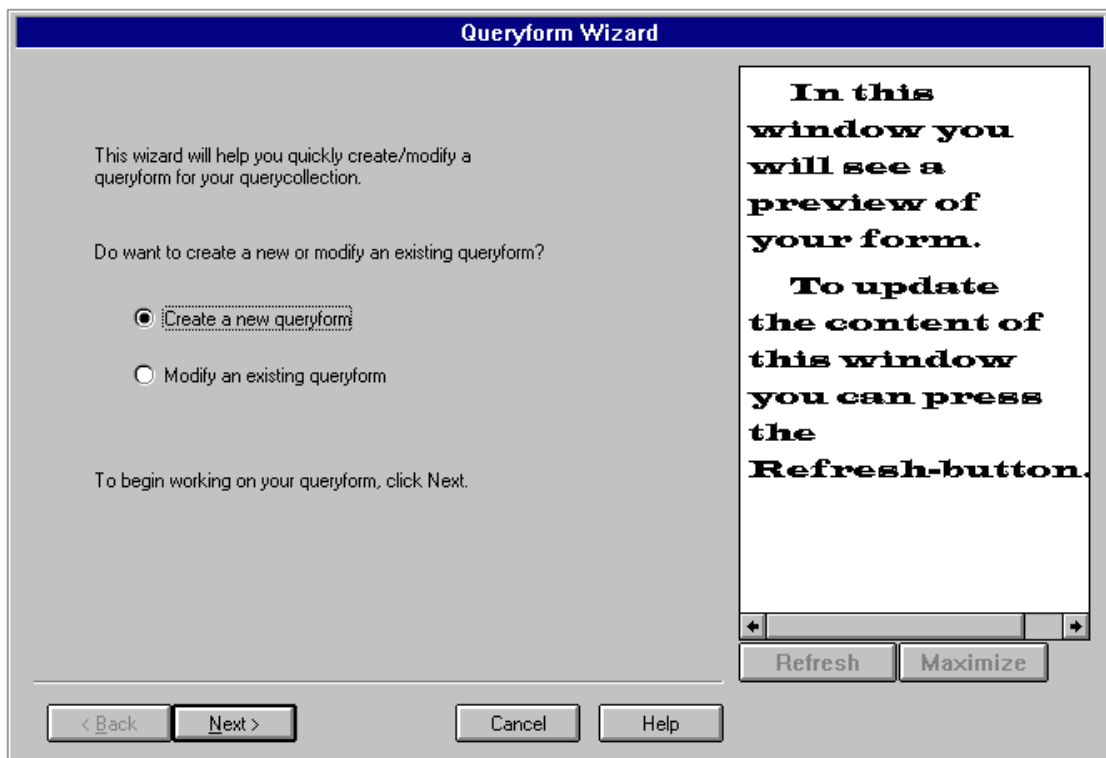


Bild 1

6.5.3.2 Titel und Beschreibung des Formulares

Hier wird nach dem Titel des Formulares gefragt und nach einer kurzen Beschreibung. Diese Beschreibung soll zukünftigen Benutzern als Hinweis dienen, wie diese Querycollection genutzt werden kann, und welche Informationen der Benutzer erhalten kann.

Auf dieser Seite besteht jetzt die Möglichkeit, durch Drücken des Navigationselementes „Refresh“ sich einen Überblick über das vorläufige Aussehen des Formulares zu machen.

Der Titel und die Beschriftung des Formulares können auch später jederzeit wieder modifiziert werden, ohne die Eingabefunktionalität zu beeinflussen.

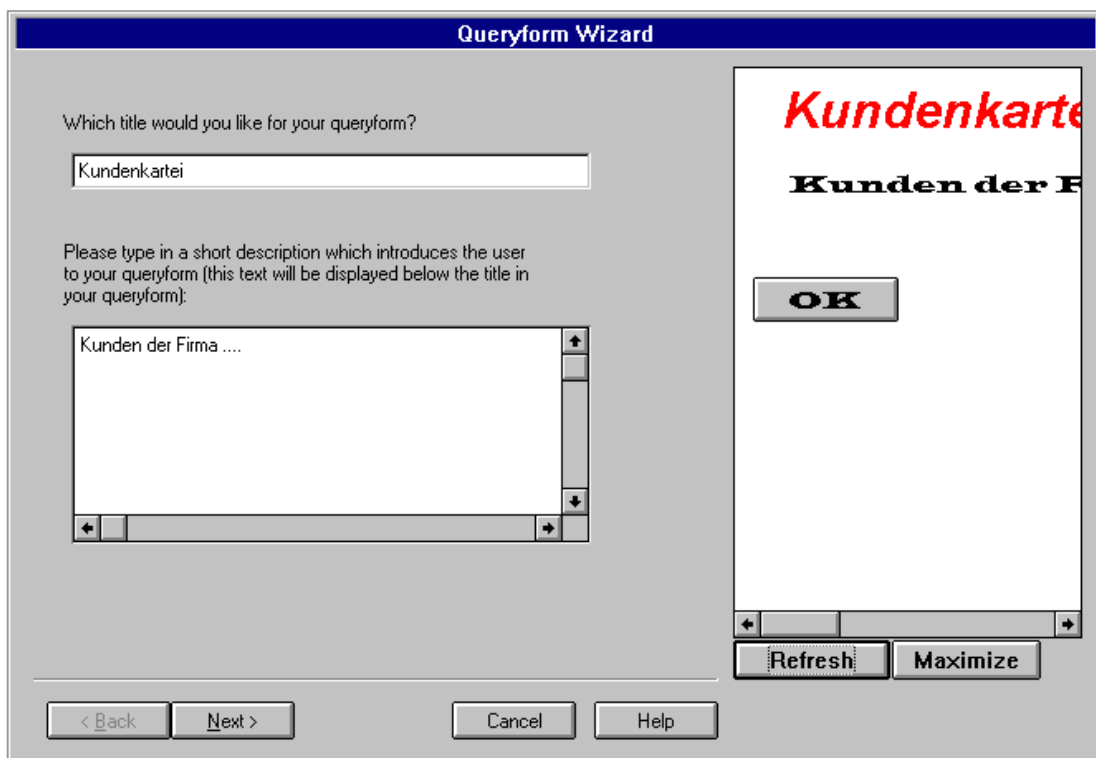


Bild 2

6.5.3.3 Bezeichnung der Metainformation

Auf der nächsten Seite besteht die Möglichkeit, die einzelnen Metainformationen genauer zu spezifizieren: Für jedes gewünschte Eingabefeld muß ein Name vergeben werden. Die Länge und das Aussehen dieses Namens ist beliebig. Als einzige Einschränkung gilt, daß für jeden Namen eine eigene Zeile verwendet werden muß.

Hinweis:

Es werden hier nur die Namen und die Reihenfolge der Eingabefelder festgelegt, die Art jedes einzelnen Eingabefeldes wird erst auf den weiteren Bildschirmseiten bestimmt.

Defaultmäßig sind alle Eingabefelder auf eine einzeilige Texteingabe voreingestellt. Es besteht jederzeit die Möglichkeit nachträglich Eingabefelder umzubenennen, in ihrer Position zu verschieben, zu löschen, oder zusätzliche Eingabefelder zu definieren. Auch wenn man schon alle Eingabefelder genau definiert hat, besteht immer die Möglichkeit, durch Drücken des Navigationselementes „Back“ auf diese Seite zu gelangen, und die Eingabefelder erneut zu bearbeiten.

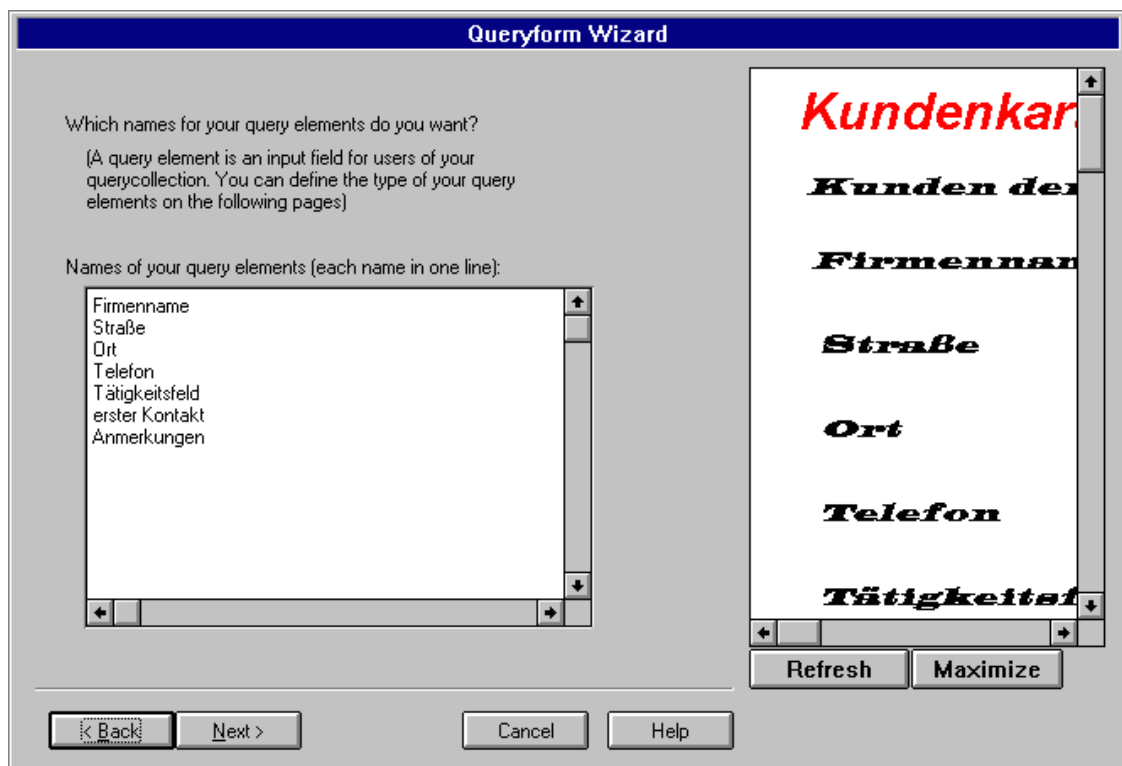


Bild 3

6.5.3.4 Arten von Eingabefeldern

Auf den weiteren Seiten können die einzelnen Eingabefelder genauer festgelegt werden. Es kann zwischen 3 verschiedenen Eingabemöglichkeiten unterschieden werden (Selbstverständlich kann diese Auswahl nachträglich wieder verändert werden):

Textfeld ("Text field")

Es kann die Größe des Textfeldes in Zeichen pro Zeile und Zeichen pro Spalte eingestellt werden. Wenn die Zeilengröße auf 1 ist, dann erhält man ein einzeiliges Eingabefeld, wenn sie größer als 1 ist, ein zweidimensionales Eingabefeld, ähnlich einem einfachen Editor.

Auch hier besteht die Möglichkeit, den Namen des Eingabeelementes nachträglich zu modifizieren.

Bei der Angabe der Breite des Textfeldes ist darauf zu achten, daß die Anzahl der Zeichen pro Zeile die Zahl 80 nicht überschreiten soll, da bei einer geringen Auflösung des Bildschirmes nicht mehr das ganze Textfeld sichtbar wäre.

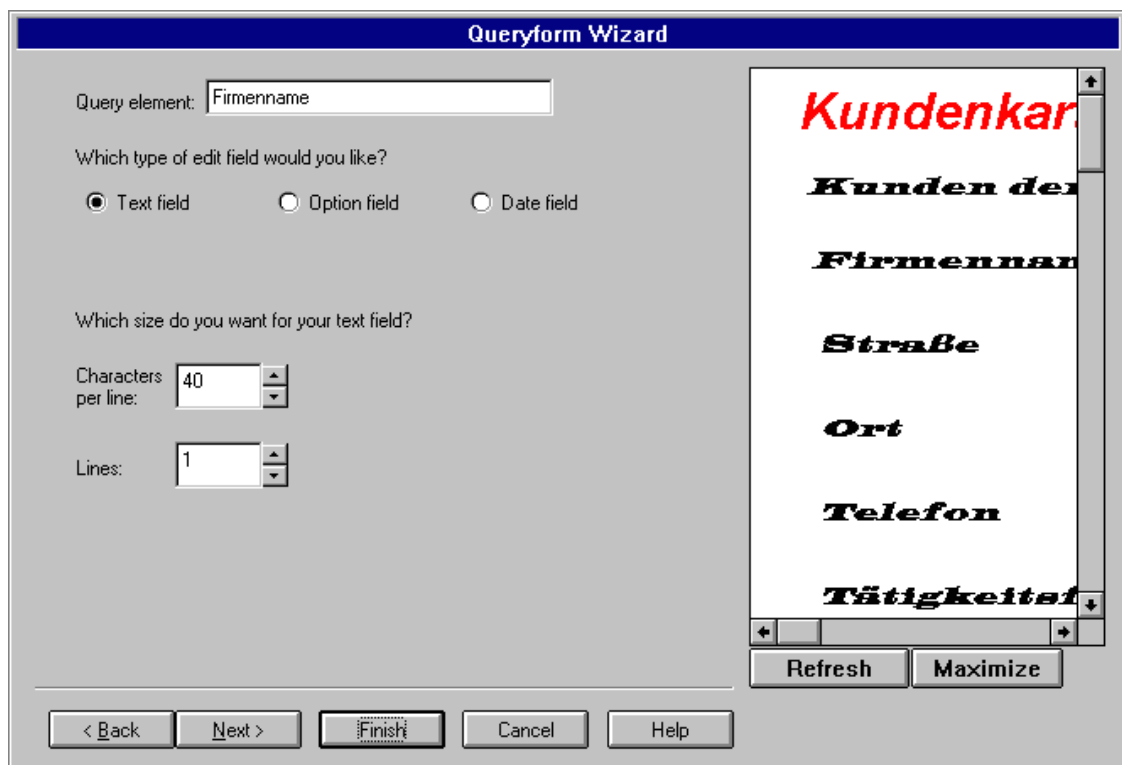


Bild 4

Auswahlfeld ("Option field")

Die einzelnen Auswahlmöglichkeiten werden wieder zeilenweise eingegeben. Bei den Auswahlfeldern kann man zwischen 2 Arten unterscheiden: Es darf immer nur eine einzige Option gewählt werden, oder es dürfen auch mehrere Optionen gewählt werden. Diese Unterscheidung kann durch "Only one option" oder "Several options" getroffen werden.

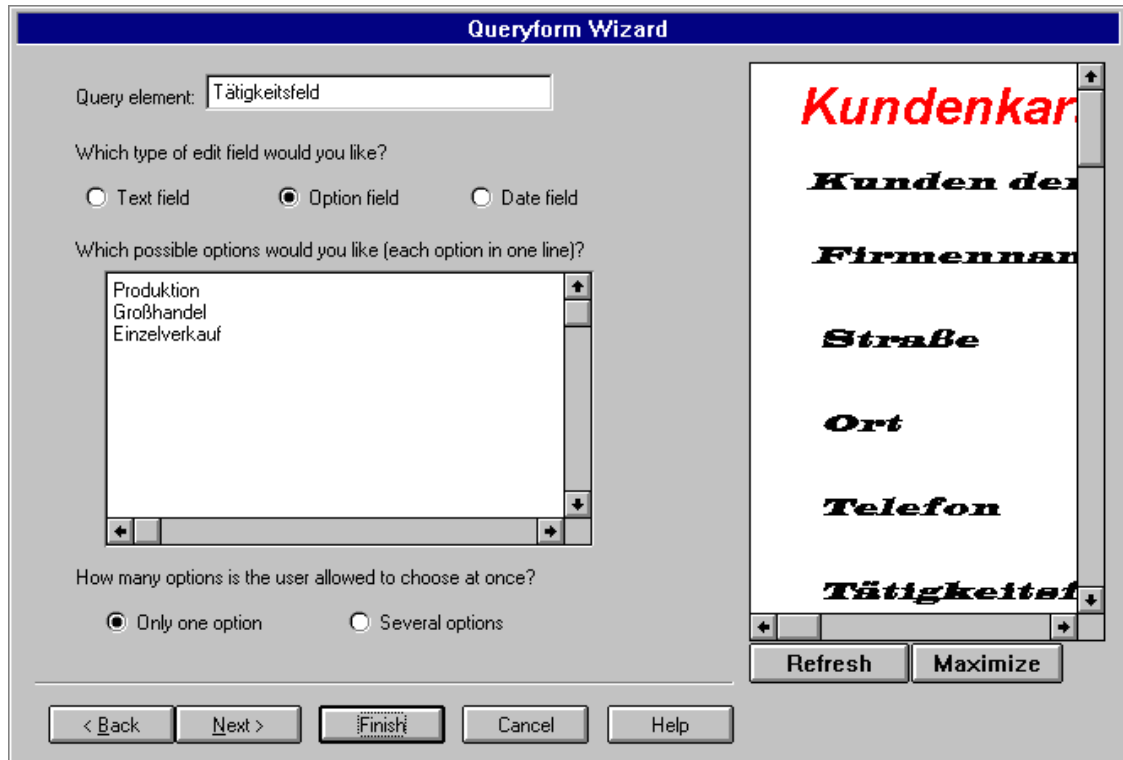


Bild 5

Datumsfeld ("Date field")

Es besteht die Möglichkeit, nur die Jahreszahl, oder Jahreszahl und Monat oder auch Jahreszahl, Monat und Tag auszuwählen.

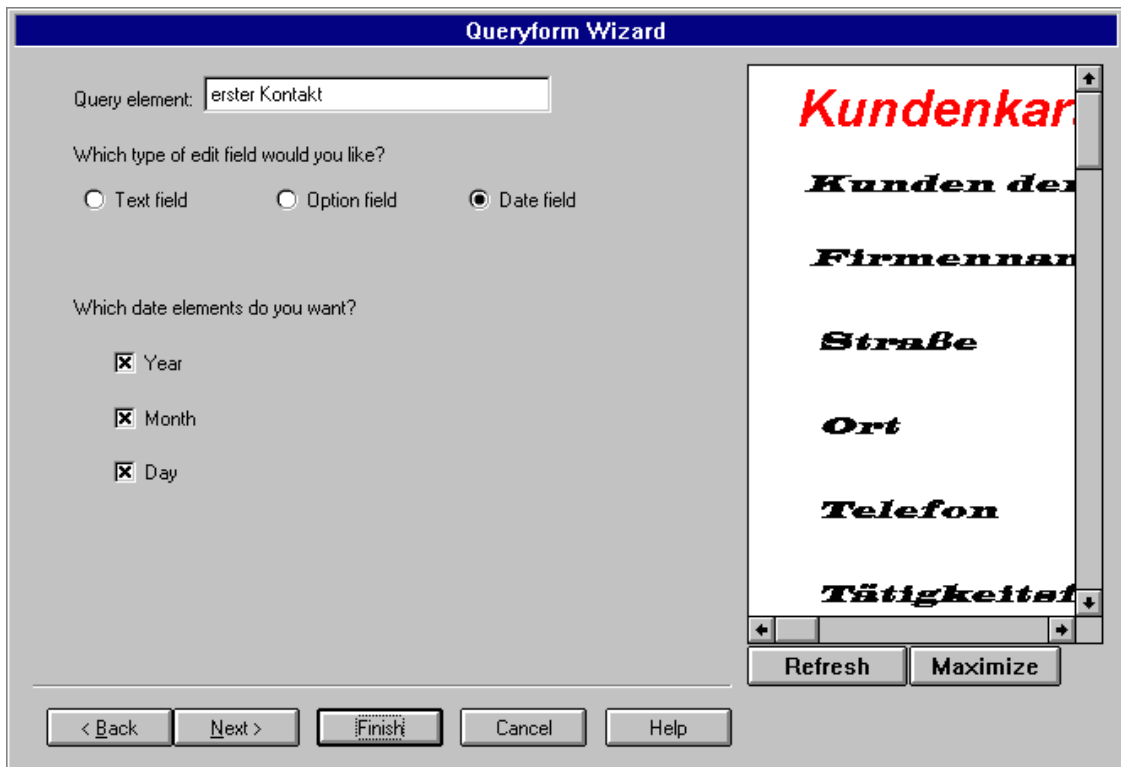


Bild 6

6.5.3.5 Aussehen des Formulars

Nachdem für alle Eingabefelder die Eingabeart festgelegt wurde, können noch einige Angaben über das Aussehen des Formulars eingegeben werden:

- Textfarbe: Die Farbe, in der der Text des Formulars erscheint.
- Hintergrundfarbe: Die Farbe, in der der Hintergrund des Formulars dargestellt wird. Es ist sehr wichtig, darauf zu achten, daß der Kontrast zwischen Text- und Hintergrundfarbe groß ist, um eine gute Lesbarkeit zu ermöglichen.
- Ausrichtung des Textes: Die Namen der Eingabefelder können links, zentriert oder rechts sein.
- Schriftstil: Beim Schriftstil kann man zwischen normal, dick, kursiv und dick-kursiv wählen.

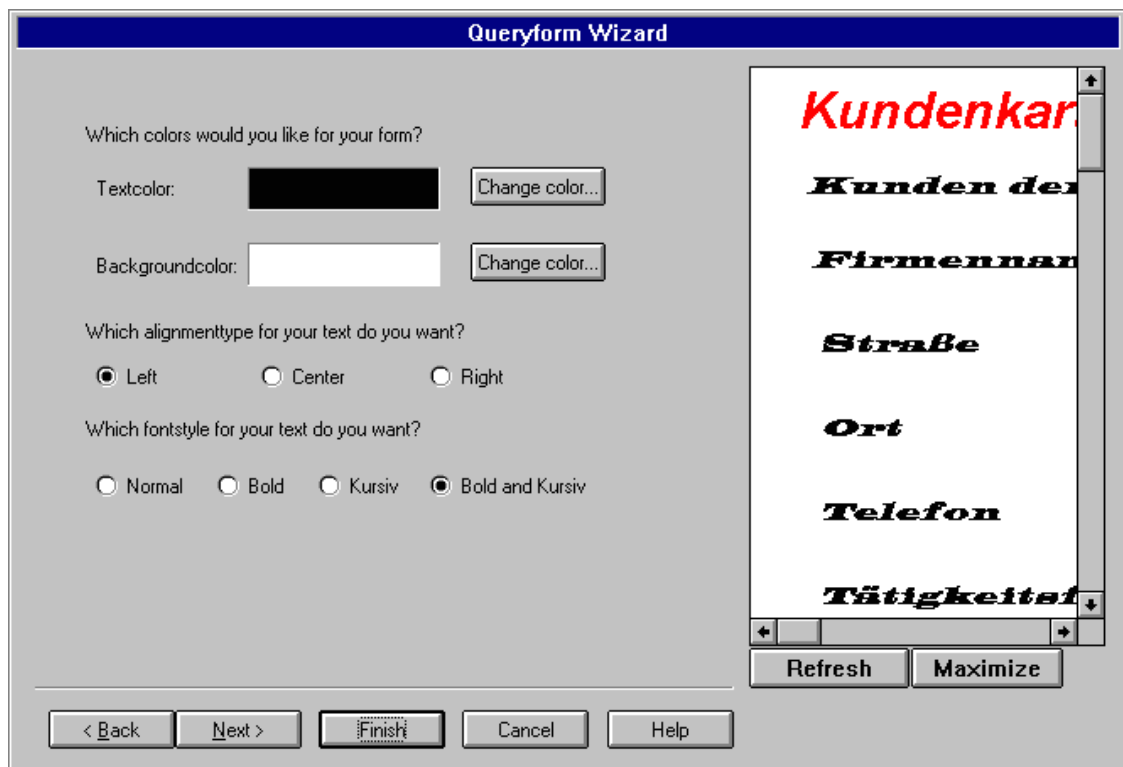


Bild 7

6.5.3.6 Möglichkeit zur Speicherung des erstellten Formulars für eine spätere Weiterbearbeitung

Auf der letzten Seite des Queryform Wizard besteht noch die Möglichkeit, das soeben erzeugte Formular lokal auf dem Computer zu speichern, um es später weiterbearbeiten zu können.

Durch Drücken des Navigationselementes „Save queryform...“ erscheint eine Standard-Dialogbox, die nach dem Dateinamen fragt, unter dem das Layout des erstellten Formulars gespeichert werden soll.

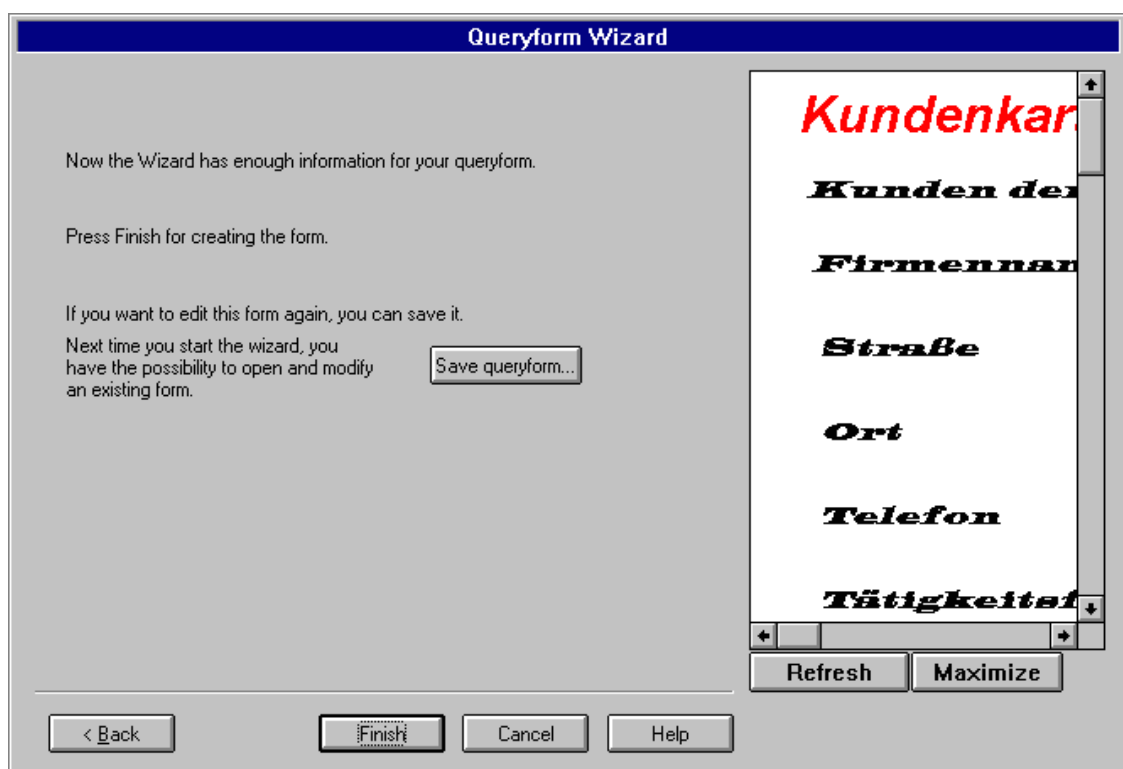
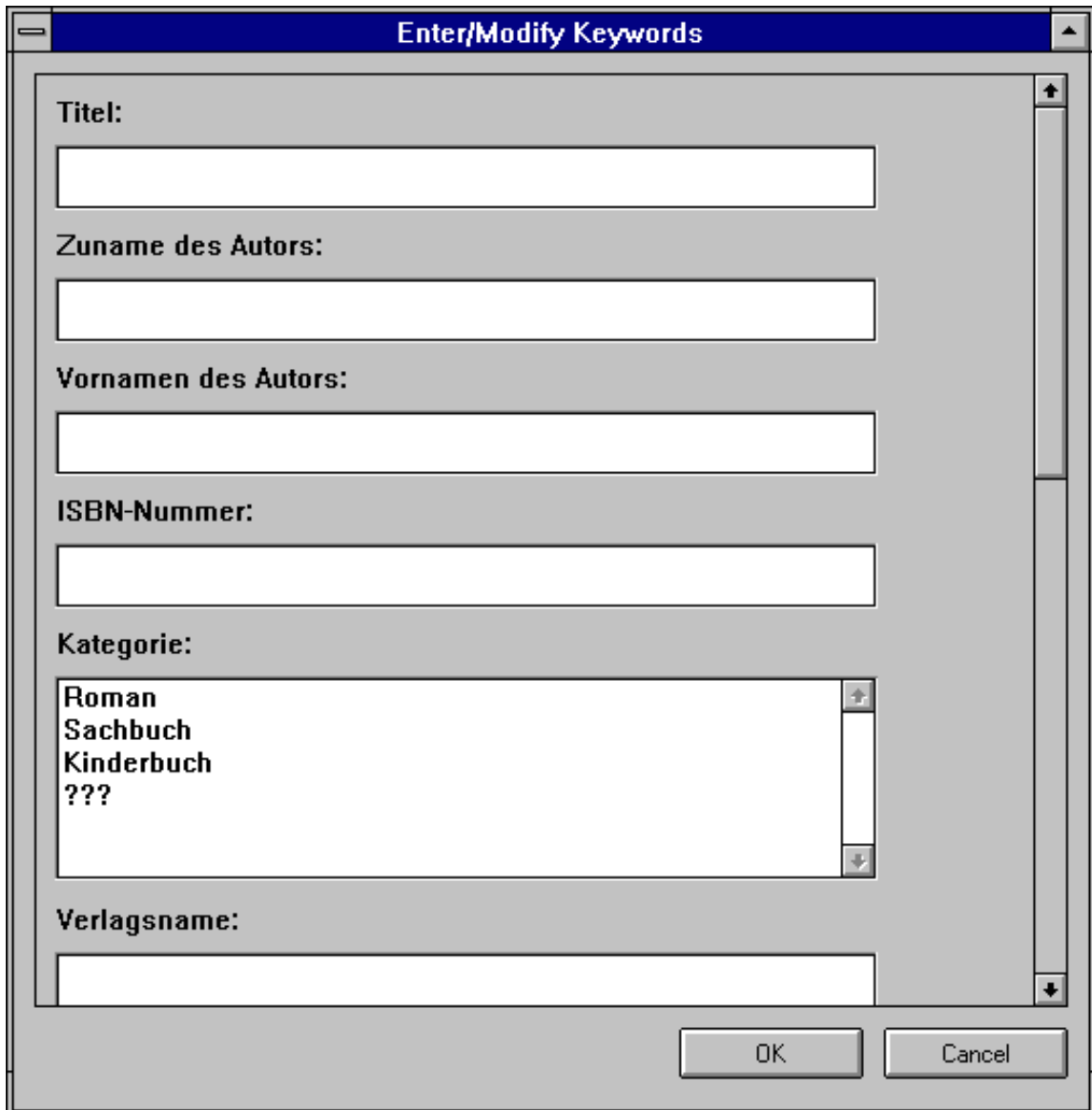


Bild 8

6.6 Eingabe der Metainformation in Amadeus

Um die jeweilige Metainformation eines neuen Dokumentes einzugeben, oder diese Metainformation mit Amadeus zu modifizieren, erscheint ein Eingabeformular, mit dem diese Informationen editiert werden können. Das Aussehen dieses Eingabeformulares richtet sich nach den Angaben bei der Erstellung des Formulars im Queryform-Wizard.



The image shows a dialog box titled "Enter/Modify Keywords". It contains the following fields:

- Titel:** A text input field.
- Zuname des Autors:** A text input field.
- Vornamen des Autors:** A text input field.
- ISBN-Nummer:** A text input field.
- Kategorie:** A list box with the following items: Roman, Sachbuch, Kinderbuch, and ???.
- Verlagsname:** A text input field.

At the bottom right of the dialog, there are two buttons: "OK" and "Cancel".

Bild 9

6.7 Suche und Suchergebnisse

Durch Betreten einer Querycollection wird automatisch das für diese Collection entsprechende Suchformular angezeigt. Durch Ausfüllen dieses Suchformulars und Abschicken an den Hyperwave-Server, wird ein CGI-Script aufgerufen, welches das Suchformular auswertet: Die Eingaben in das Formular werden zu einem "Query-String" zusammengesetzt und dem Hyperwave-Tool Hwinfo übergeben. Die Suchergebnisse von Hwinfo werden aufbereitet und als HTML-Dokument an den Client geschickt.

Kundenkartei

Kunden der Firma

Firmenname

Straße

Ort

Telefon

Tätigkeitsfeld

erster Kontakt (1996 not 96)

Anmerkungen

Bild 10

6.8 Programmierung

6.8.1 Queryform-Wizard

Der Queryform-Wizard wurde speziell für den Hyperwave-Client Amadeus entwickelt. Als Programmiersprache wurde C++ ([Lippman 91] [Stroustrup 91]) und als Entwicklungsumgebung Microsoft Visual C++ ([Kruglinski 93]) mit der Microsoft Foundation Class 4.0 verwendet.

Der HTML-Viewer im Wizard ist eine vereinfachte Version des HTML-Viewers von Amadeus.

6.8.2 Enter/Modify Keywords in Amadeus

Eine einfache Eingabemaske, um beim Einfügen von neuen Dokumenten in die Querycollection über Amadeus die Metainformation eingeben zu können. Die Eingabefelder des Formulars passen sich automatisch den Vorgaben durch den Queryform-Wizard an. Als Programmiersprache wurde ebenfalls Microsoft Visual C++ mit der Microsoft Foundation Class 4.0 verwendet.

6.8.3 CGI-Script für die Suche

Um die Suche am Hyperwave-Server durchzuführen wird ein CGI-Script ([December 95] [Maurer 96] [Liu 94]) verwendet, das in Perl ([Till 95] [Schwartz 94]) programmiert ist, und unter Verwendung des Hyperwave Tools hwinfo die Suchergebnisse an den Benutzer als HTML-Seite zurückliefert.

7 REFERENZEN

- [Agosti 96] AGOSTI, MARISTELLA: *Information Retrieval and Hypertext*; Kluwer Academic Publishers 1996
- [Apple 87] *Human Interface Guidelines: The Apple Desktop Interface*; Addison-Wesely Publishing Company 1987
- [Bass 91] BASS, LEN; COUTAZ JOELLE: *Developing Software for the User Interface*; Addison-Wesely Publishing Company 1991
- [Berk 91] BERK, EMILY; DEVLIN, JOSEPH: *Hypertext/Hypermedia Handbook*, McGraw-Hill Verlag 1991
- [Chandler 95] CHANDLER, DAVID M.: *Running a Perfect Web Site*, Que Corporation 1995
- [Comer 88] COMER, DOUGLAS: *Internetworking with TCP/IP*, Prentice Hall 1988
- [Dalitz 95] DALITZ, WOLFGANG; HEYER, GERNOT; *Hyperwave Das Internet-Informationssystem der 2. Generation*; dpunkt Verlag für digitale Technologie Heidelberg 1995
- [Dalitz 96] DALITZ, WOLFGANG; HEYER, GERNOT; *Hyperwave The New Generation Internet System Based on Hyper-G Technology*; dpunkt Verlag für digitale Technologie Heidelberg 1996
- [December 95] DECEMBER, JOHN; GINSBURG, MARK: *HTML and CGI Unleashed*, Sams.net 1995
- [Dietinger 95] DIETINGER, THOMAS; *Object-oriented implementation of a multi-protocol hyper-G client for Ms-Windows*; Diplomarbeit (1995) in Telematik am Institut für Informationsverarbeitung und computergestützte neue Medien, Technische Universität Graz
- [Flohr 95] FLOHR, UDO; *Hyper-G Organizes the Web*; BYTE November 1995
- [Flohr 96] FLOHR, UDO; *Hyper-G ohne Geheimnis*; Verlag Heinz Heise 1996
- [Foly 90] FOLEY, JAMES D., VAN DAM, ANDRIES: *Computer Graphics - Principles and Practice*; Addison-Wesely Publishing Company 1990
- [Frakes 92] FRAKES, WILLIAM: *Information Retrieval - Data Structures & Algorithms*; Prentice Hall 1992
- [Graham 95] GRAHAM, IAN S.: *The HTML Sourcebook*, Wiley Verlag 1995

- [Heller 89] HELLER, EVA: *Wie Farben wirken*; Rowohlt Verlag 1989
- [Houghton 84] HOUGHTON, BERNARD; CONVEY, JOHN; *Online Information Retrieval Systems*; Clive Bingley 1984
- [Jones 95] JONES, RUSS; NYE, ADRIAN; MERZ, THOMAS [BEARB.]: *HTML und das World Wide Web*, O'Reilly, Internat. Thomson-Verlag 1995
- [Jorge 96] JORGE, STEFFEN: *HTML 3.2 – Referenzhandbuch*, Sybex Verlag 1996
- [Krol 94] KROL, ED: *The Whole Internet User's Guide & Catalog*, O'Reilly & Associates, Inc 1994
- [Kruglinski 93] KRUGLINSKI, DAVID J.: *Inside Visual C++*, Microsoft Press 1993
- [Kuhlen 80] KUHLEN, RAINER: *Datenbasen, Datenbanken, Netzwerke. Praxis des Information Retrievals*; K.G. Saur 1980
- [Kyas 94] KYAS, OTHMAR: *Internet: Zugang, Utilities, Nutzung*, Datacom-Verlag 1994
- [Lancaster 78] LANCASTER, F.W.: *Toward Paperless Information Systems*; Academic Press 1978
- [Laurel 91] LAUREL, BRENDA: *Computers as Theatre*; Addison-Wesely Publishing Company 1991
- [Lippman 91] LIPPMAN, STANLEY B.: *The C++ Primer*, Addison-Wesley 1991
- [Liu 94] LIU, CRICKET; PEEK, JENNY; JONES, RUSS: *Managing Internet Information Services*, O'Reilly Verlag 1994
- [Mary 96] MARY E.; MORRIS, S.; HINRICHS, RANDY J.: *Web Page Design*, SunSoft Press 1996
- [Maurer 96] MAURER, RAINER: *HTML und CGI-Programmierung*, dpunkt Verlag für digitale Technologie 1996
- [MaurerH 96] MAURER, HERMANN; *Hyper-G now Hyperwave*; Addison Weseley 1996
- [Micro 95] *The Windows Interface Guidelines for Software Design*; Microsoft Press 1995
- [Mresse 84] MRESSE, M.: *Information Retrieval - Eine Einführung*; B.G. Teubner 1984
- [Niederst 96] NIEDERST, J.; FREDDMAN, E.: *Designing for the Web*; O'Reilly 1996

- [Preece 94] PREECE, JENNY: *Human Computer Interaction*; Addison-Wesely Publishing Company 1994
- [Raitt 84] RAITT, DAVID: *Introduction to Online Information Systems*; Oxford and New Jersey 1984
- [Rowley 87] ROWLEY, JENNIFER: *Organising Knowledge - An Introduction to Information Retrieval*; Gower Verlag 1987
- [Salton 68] SALTON, GERARD: *Automatic Information Organization and Retrieval*; McGraw-Hill 1968
- [Schwartz 94] SCHWARTZ, RANDAL L.: *Learning Perl*, O'Reilly 1994
- [Sengstack 96] SENGSTACK, JEFF; *Hyper-G: The Next Generation Web Server?*; Newmedia Mai 1996
- [Shneider 92] SHNEIDERMAN, BEN: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*; Addison-Wesely Publishing Company 1992
- [Stroustrup 91] STROUSTRUP, BJARN: *The C++ Programming Language*, Addison-Wesley 1991
- [Till 95] TILL, DAVID: *Perl in 21 Days*, Sams Publishing 1995
- [Wilson 95] WILSON, STEPHEN: *World Wide Web Design Guide*, Hayden Books 1995
- [Witten 94] WITTEN, IAN H.; MOFFAT, ALISTAIR; BELL, TIMOTHY C.: *Managing Gigabytes - Compressing and Indexing Documents and Images*; Verlag Van Nostrand Reinhold 1994