

Seminar Multimedia und Electronic Publishing



WS 1996/97

**Universität Karlsruhe
Institut für Betriebs- und Dialogsysteme
Abteilung Dialogsysteme und graphische Datenverarbeitung
Prof. Dr. A. Schmitt**

Khaldoun Ateyeh, Günther Augustin, Matthias Baas, Olfa Karboul, Alexander Krieg, Karsten Kutschera, Michael Meichel, Boris Moser, Jürgen Moßgraber, Eduard Niess, Heiko Roth, Andreas Steinbuch, Jewgenij Tarnorutskiy

**Betreuer:
Michael Fautz
Peter Oel
Kurt Saar**

In den letzten Jahren hat die Zahl der auf elektronischem Weg veröffentlichten Dokumente stark zugenommen. Eng verknüpft mit elektronischen Dokumenten ist auch der Bereich Multimedia. Im Rahmen dieses Seminars wurden Themen aus den Bereichen Multimedia und elektronische Veröffentlichung (Electronic Publishing) besprochen. Inhaltlich ist das Seminar in drei Themengebiete aufgeteilt:

Der erste Teil schließt an das Seminar *Multimedia-Datenformate* an, das 1994 ebenfalls am Institut für Betriebs- und Dialogsysteme, Lehrstuhl Prof. Dr. A. Schmitt, stattfand. Nach einem Überblick zum Thema Multimedia werden neuere Datenformate und Programmiersprachen aus dem Multimedia-Bereich untersucht (Kapitel 1 - 4).

Der zweite Teil befaßt sich mit der Thematik des elektronischen Veröffentlichens. Nach einem Überblick und der Betrachtung einiger Layout- und Design-Richtlinien (Kapitel 5 und 6) werden drei Autorensysteme zur Erstellung von Multimedia-Anwendungen untersucht, und zwar Asymetrix Toolbook, APE Mediastylers und Macromedia Authorware (Kapitel 7 - 9). Das daran anschließende Kapitel 10 gibt einen Überblick über die aktuellen CD-ROM-Typen. Der zweite Teil schließt mit einer Betrachtung der rechtlichen Aspekte des elektronischen Veröffentlichens (Kapitel 11).

Der dritte Teil des Seminars beinhaltet die Darstellung einiger multimedialer Anwendungen. Hier werden der Bereich des computerunterstützten Lernens (Computer Based Training) und Kiosksysteme betrachtet (Kapitel 12 und 13).

Auf diesem Weg sei noch einmal allen Seminar-Teilnehmern für ihre Mitarbeit gedankt, insbesondere Matthias Baas, der die Zusammenstellung der Ergebnisse tatkräftig unterstützt hat.

Diese Ausarbeitung ist auch im WWW verfügbar:
<http://i31www.ira.uka.de/docs/mm+ep/>

Inhaltsverzeichnis

1	Multimedia: Ein Überblick	1
1.1	Übersicht	1
1.2	Einleitung	1
1.3	Was ist Multimedia? - Begriffsklärung	1
1.3.1	Wortimmanente Definition	1
1.3.2	Ziele von Multimedia-Systemen	4
1.4	Multimedia-Mittel, -Formate und -Hardware	4
1.4.1	Multimedia-Mittel	5
1.4.2	Datenformate und Kompressionsverfahren	6
1.4.3	Multimedia-Hardware	7
1.5	Entwicklung multimedialer Anwendungen	9
1.6	Nutzung von Multimedia und Electronic Publishing	10
1.7	Ausblick	11
1.8	Literatur	11
2	PNG	13
2.1	Übersicht	13
2.2	Einleitung	13
2.3	Aufbau einer PNG-Datei	15
2.3.1	Signatur	15
2.3.2	Chunks	16
2.3.3	Unverzichtbare Chunks	17
2.3.4	Zusatzchunks	19
2.4	Kompression	20
2.4.1	Verwendeter Algorithmus	20
2.4.2	Filter	20
2.5	Interlacing	21
2.5.1	Was ist Interlacing	21
2.5.2	Interlacing in GIF-Bildern	23
2.5.3	Interlacing in PNG-Bildern	23
2.6	Transparenz	24
2.6.1	Was ist Alpha?	24
2.6.2	Einfache Transparenz	24
2.6.3	Alpha-Kanal	25
2.7	Farbwiedergabe	25
2.8	Zusammenfassung	26
2.9	Literatur	27

3	Virtual Reality Modeling Language (VRML): Interaktive 3D-Dokumente	29
3.1	Übersicht	29
3.2	Einleitung	29
3.3	Entstehungsgeschichte	29
3.3.1	VRML 1.0	29
3.3.2	VRML 2.0 (Moving Worlds)	30
3.4	Wozu braucht man VRML ?	30
3.5	Was kann VRML ?	30
3.5.1	Die Möglichkeiten von VRML 1.0	31
3.5.2	Die Möglichkeiten von VRML 2.0	31
3.6	Spezifikation	32
3.6.1	Grundlagen der Sprache	32
3.6.2	Das Koordinatensystem	33
3.6.3	Die Struktur des Szenengraphen	33
3.6.4	Felder (<i>fields</i>)	34
3.6.5	Events	36
3.6.6	Nodes (Knoten)	37
3.6.7	Erweiterbarkeit (Prototypen)	52
3.6.8	Verwendung von Scripts	54
3.7	Anwendungen von VRML	54
3.7.1	Architektur	55
3.7.2	Chemie	55
3.7.3	Virtuelle Welten	55
3.7.4	Mathematik	55
3.8	Literatur	55
4	Java und Java Applets	57
4.1	Übersicht	57
4.2	Einleitung	57
4.3	Spracheigenschaften	57
4.3.1	Primitive Datentypen	57
4.3.2	Höhere Datentypen	58
4.3.3	Speicherverwaltung	58
4.3.4	Fehler- und Ausnahmebehandlung	58
4.3.5	Objektorientiertes Programmieren	58
4.3.6	Packages	59
4.4	Java-Ausführungsumgebung	59
4.5	Die Schnittstelle zwischen HTML und Java-Applets	60
4.6	Applet-Ausführung	61
4.7	Applets und Sicherheit	62
4.8	Ausblick	62
4.9	Literatur	62
5	Electronic Publishing: Ein Überblick	63
5.1	Übersicht	63
5.2	Einleitung	63
5.3	Realisierungsmöglichkeiten	64
5.4	Definitionen	64
5.4.1	Informationsobjekte	64
5.4.2	Beziehungen	65
5.5	Entwicklungsprozeß	66
5.5.1	Entwicklungsphasen	66
5.5.2	Entwicklungswerkzeuge	68
5.6	Elektronisches Publizieren	68

5.6.1	Autorensysteme für die elektronische Offline-Publikation	68
5.6.2	Autorensysteme für die elektronische Online-Publikation	69
5.7	Hypertext/Hypermedia	70
5.7.1	Motivation für die Entwicklung	70
5.7.2	Charakteristische Merkmale	70
5.8	Bewertung	70
5.9	Literatur	70
6	Layout-/Design-Richtlinien für elektronische Dokumente	73
6.1	Übersicht	73
6.2	Grundlegendes	73
6.2.1	Ausgangssituation	73
6.2.2	Zielgruppe	73
6.2.3	Ziel	73
6.3	Allgemeine Regeln	74
6.3.1	Sprache	74
6.3.2	Gestaltgesetze	76
6.4	Typografie	77
6.4.1	Schriftcharakterisierung	77
6.4.2	Schriftwahl	78
6.4.3	Überschriften	79
6.4.4	Satzarten	80
6.5	Layout	80
6.5.1	Corporate Design	82
6.6	Elektronische Dokumente	82
6.6.1	Informationelle Einheit (Seite)	82
6.6.2	Nicht-Linearität	83
6.6.3	Orientierungs- und Navigationshilfen	84
6.6.4	Skripten	84
6.7	World Wide Web	85
6.8	Literatur	85
7	Multimedia ToolBook	87
7.1	Übersicht	87
7.2	Klasseneinteilung der Autorensysteme	87
7.2.1	Frame-basierte Autorensysteme	87
7.2.2	Timeline-basierte Autorensysteme	87
7.2.3	Flowchart-basierte Autorensysteme	87
7.3	Grundlegende Konzepte von Multimedia ToolBook	88
7.3.1	frame-basiert	88
7.3.2	ereignisorientiert	88
7.3.3	Objekthierarchie	88
7.3.4	skript-orientiert	89
7.3.5	Autor-/Lesermodus	89
7.4	verfügbare Objekte	89
7.4.1	graphische Objekte	89
7.4.2	Buttons	89
7.4.3	Texte	90
7.4.4	Hotwords	90
7.4.5	Listen und Comboboxen	90
7.4.6	Menüs	90
7.4.7	Bilder	90
7.4.8	Gruppen	91
7.4.9	Beispiel	91

7.5	OpenScript	91
7.5.1	Kommentare	92
7.5.2	ToolBook-Objekte	92
7.5.3	Ereignisse und Handler	92
7.5.4	Funktionen	92
7.5.5	Variablen	93
7.5.6	Kontrollstrukturen	93
7.5.7	Beispiel	93
7.6	weiterführende Möglichkeiten	93
7.7	Verfügbarkeit	94
7.7.1	Preise	94
7.7.2	Hardwareanforderungen	94
7.8	Zusammenfassung	95
7.8.1	Vorteile	95
7.8.2	Nachteile	95
7.8.3	Einsatzgebiete	96
7.8.4	Fazit	96
7.9	Einige WWW-Adressen	96
7.10	Literatur	96
8	APE Media Styler	97
8.1	Übersicht	97
8.2	Einleitung	97
8.3	Leistungsumfang	97
8.3.1	Die Versionen	97
8.3.2	Unterstützte Datenformate	98
8.3.3	Unterstützte Hardware	98
8.3.4	Übersicht über die Schnittstellen	98
8.4	Benutzungskonzept	99
8.4.1	Seitenbasierte Autorensysteme	99
8.4.2	Benutzerführung	99
8.5	Die Komponenten einer Media Styler Applikation	99
8.5.1	Übersicht	99
8.5.2	Fenster (<i>windows</i>)	100
8.5.3	Auslöseknöpfe (<i>pushbuttons</i>)	101
8.5.4	Textobjekte (<i>text object</i>)	101
8.5.5	Medienbox (<i>media box</i>)	101
8.5.6	Feldobjekt (<i>field object</i>)	101
8.6	Komponenten zur Erstellung einer Media Styler Applikation	105
8.6.1	Toolkit	105
8.6.2	Bildverarbeitung (<i>image processing</i>)	105
8.6.3	Bewegungspfad für Objekte (<i>move-path feature</i>)	105
8.6.4	Dia-Show-Editor (<i>slide show editor</i>)	105
8.6.5	Benutzer Pop-Up-Menü (<i>user pop-up menu</i>)	105
8.7	Bewertung	106
8.7.1	Vorteile des Konzepts	106
8.7.2	Nachteile des Konzepts	106
8.7.3	Arbeiten in der Praxis	107
8.7.4	Fazit	107
8.8	Literatur	107

9	Authorware	109
9.1	Übersicht	109
9.1.1	Einleitung	109
9.1.2	Einordnung	109
9.2	Anwendungsgebiete für Authorware	110
9.3	Plattformverfügbarkeit	110
9.4	Arbeitsoberfläche	111
9.4.1	Darstellungs-Icon	112
9.4.2	Bewegungs-Icon	112
9.4.3	Lösche-Icon	112
9.4.4	Warte-Icon	112
9.4.5	Navigations-Icon	112
9.4.6	Rahmen-Icon	112
9.4.7	Entscheidungs-Icon	112
9.4.8	Interaktions-Icon	113
9.4.9	Berechnungs-Icon	113
9.4.10	Gruppierungs-Icon	113
9.4.11	Digital-Movie-Icon	113
9.4.12	Ton-Icon	113
9.4.13	Video-Icon	113
9.4.14	Start- und Stopflagge	113
9.4.15	Farbpalette	113
9.5	Variablen und Funktionen	114
9.6	Kriterienkatalog	115
9.6.1	Verfügbarkeit	116
9.6.2	Entwicklung	116
9.7	Fazit	118
9.8	Weiterführende Informationsquellen	118
9.9	Literatur	118
10	CD-ROM	121
10.1	Übersicht	121
10.2	CD-ROM Hardware	121
10.2.1	Aufbau einer CD	121
10.2.2	EFM-Modulation	123
10.2.3	Pressen und Brennen einer CD	124
10.3	CD-ROM Standards und Formate	124
10.3.1	CD-DA	124
10.3.2	CD-ROM	127
10.3.3	Mixed-Mode-CD	127
10.3.4	CD-ROM-XA	128
10.3.5	CD-I	128
10.3.6	CD-I-Ready	128
10.3.7	CD-I-Bridge und Video-CD	128
10.3.8	CD-MO und CD-WO bzw. CD-R	128
10.3.9	Photo-CD	129
10.3.10	Enhanced-Music-CD	129
10.3.11	Filesysteme	129
10.3.12	Hybrid-CD	130
10.4	DVD	130
10.5	Neuer Ansatz	130
10.6	Recycling von CDs	131
10.7	Literatur	131

11 Rechtliche Aspekte elektronischer Veröffentlichung	133
11.1 Übersicht	133
11.2 Einleitung	133
11.3 Multimedia	134
11.3.1 Was ist Multimedia - aus rechtlicher Perspektive	134
11.4 Das Urheberrecht	135
11.4.1 Gegenstand und Schutzbereich des Urheberrechts	135
11.4.2 Was ist alles geschützt?	137
11.4.3 Was ist ein Werk?	137
11.5 Urheberrecht und digitale Werknutzung	137
11.6 Der digitale Herausgeber / Werkbearbeitungen	138
11.7 Übernahme und Veränderungen bestehender Werke	138
11.7.1 Die Bearbeitung und Umgestaltung vorbestehender Werke	138
11.7.2 Schutz vor Entstellungen	138
11.7.3 Zitat	139
11.7.4 Das Plagiat	139
11.7.5 Die Parodie	139
11.7.6 Gemeinfreie Werke	140
11.8 Erwerb und Vergütung der Nutzungsrechte	140
11.8.1 Verwertungsgesellschaften	141
11.9 Hilfen bei der Vertragsgestaltung bei Multimediaprojekten	141
11.10 Internet und Urheberrecht	142
11.11 Einige WWW-Adressen zum Thema Recht	143
11.12 Literatur	144
12 Computer Based Training	145
12.1 Übersicht	145
12.2 Einleitung	145
12.3 CBT als Alternative	145
12.4 Varianten und Einsatzgebiete	146
12.4.1 Tutorielle Formen: Programmierte Instruktionen und Übungen	146
12.4.2 Nichttutorielle Formen: Computerunterstützte didaktische Simulationen und Spiele	147
12.5 Intelligente tutorielle Lernsysteme	148
12.6 Hypertextsysteme	149
12.7 Werkzeuge für Entwicklung von Lernsoftware	149
12.8 Möglichkeiten und Grenzen interaktiver Medien in der betrieblichen Bildung	150
12.8.1 Adressaten	150
12.8.2 Lerninhalte	150
12.8.3 Lernziele/Lerntechniken	151
12.8.4 Wirtschaftlichkeit	151
12.8.5 Lernsituation	152
12.8.6 Schlußbetrachtungen	153
12.9 Neues Lernen im Netz	153
12.10 Literatur	154
13 Kiosksysteme	155
13.1 Übersicht	155
13.2 Einleitung	155
13.2.1 Was ist ein Kiosksystem?	155
13.2.2 Charakteristiken	155
13.2.3 Klassifikation	156
13.3 Architektur und Bausteine eines Kiosksystems	156
13.4 Anwendungsbereiche	157
13.4.1 Einsatzgebiete	157

13.5	Multimediale Kiosksysteme	158
13.5.1	Was ist Multimedia	158
13.5.2	Verteilte multimediale Kiosksysteme	158
13.5.3	WWW als Kiosksystem	159
13.6	Bewertung	160
13.6.1	Vorteile eines Kiosksystem	160
13.6.2	Nachteile eines Kiosksystem	161
13.6.3	Bewertung des Fraunhofer-Instituts	162
13.7	Literatur	162



Kapitel 1

Multimedia: Ein Überblick

Michael Meichel

1.1 Übersicht

Diese Ausarbeitung gibt einem kurzen Überblick über Multimedia. Da es sich bei Multimedia um ein junges Teilgebiet der Informatik handelt, ist zu bedenken, daß die folgenden Ausführungen nur das momentane Verständnis von Multimedia widerspiegeln.

In diesem Überblick wird der Multimediabegriff unter dem informationstechnischem Aspekt erläutert und gegenüber anderen Fachgebieten abgegrenzt werden. Desweiteren werden Gliederungen von Medien aufgezeigt. Dabei erfolgt eine Unterteilung von Medien in Multimedia-Mittel, -Formate und -Hardware.

1.2 Einleitung

In unserer schnelllebigen Zeit vollzieht sich ein stetiger gesellschaftlicher und industrieller Wandel. Ein neuer Produktionsfaktor gewinnt dabei zunehmend an Bedeutung. Bei diesem Produktionsfaktor handelt es sich, obwohl immateriell, um einen essentiellen Faktor. Dieser Faktor ist die Information. Die Informationstechnik (IT) ist die Lehre vom Umgang mit Informationen.

Dem Produktionsfaktor Information, wird ein ebensogrosser Stellenwert zugemessen, wie vormalis der Dampfmaschine, die die Industrielle Revolutions einläutete. Deshalb verwundert es auch nicht, wenn mancherorts von einer IT-Revolution gesprochen wird. Die IT-Revolution soll ähnliche evolutionäre und gesellschaftliche Umwälzungen wie die industrielle Revolution bringen und diese ablösen. Es entstehen neue Berufsgruppen, die sich mit Informationen beschäftigen. Die Informationstechnik ist die Lehre vom Umgang mit Informationen.

Multimedia ist ein Teilgebiet der Informationstechnik. Deren allgemeine Zielsetzung in der (vielsichtigen) Informationsvermittlung und -darstellung liegt. Die Bedeutung von Multimedia wird unter anderem durch den Titel 'Wort des Jahres 1995' unterstrichen.

1.3 Was ist Multimedia? - Begriffsklärung

In diesem Abschnitt erfolgt eine Klärung des Multimediabegriffs. Zunächst wird untersucht, wie der Multimediabegriff in der Informationstechnik und in anderen Fachbereichen wie bspw. der Kunst gebraucht wird. Eine differenzierte Betrachtung wird in den folgenden Unterabschnitten vorgenommen.

1.3.1 Wortimmanente Definition

Ein erster Ansatz zur Klärung des Multimediabegriffs ist durch eine nähere Betrachtung des Wortes 'Multimedia' gegeben. Multimedia ist eine Wortzusammensetzung aus 'multi-' und 'media'.

Beide Wörter stammen aus dem Lateinischen. Das Präfix 'Multi-' steht für 'viel-' und 'media' für 'das in der Mitte Befindliche' oder 'Mittel'. Wenn man das allgemeine Kommunikationsschema zugrundelegt



Abbildung 1.1: Allgemeines Kommunikationsschema

(Abb. 1.1), erkennt man, daß mit Multimedia (in der Mehrzahl) Mittel gemeint sind, die zur Verständigung zwischen einem Sender und einem Empfänger dienen.

1.3.1.1 Multimedia - Begriffsabgrenzung

Multimedia ist ursprünglich kein von den Informationstechnikern geprägter Begriff. Die Kunst benutzt den Begriff Multimedia schon seit Jahrzehnten. Wenn man im Brockhaus [Bro91] nachschaut, erhält man hierzu folgende Definition:

“Multimedia: (auch Mixed Media gen.) meint die zeitgenöss. Äußerungsformen, die eine Verbindung mehrerer Kunstbereiche (...) zu einer neuen Einheit im Sinne eines Gesamtkunstwerks anstrebt, unter Einbeziehung verschiedenster techn. Medien (Photographie, Film, Tonband, Video, Laser u. a.), wobei der besondere Akzent nicht nur auf die Aufhebung der Kunstgattung, sondern auch auf eine Abhebung der Diskrepanz von Leben und Kunst gelegt wird.” [Bro91, Bro93]

Ausgangspunkt sind mehrere Kunstbereiche, die zu einem Gesamtkunstwerk unter Nutzung technischer Medien zusammengefügt werden. Technische Medien und Äußerungsformen werden dabei unterschieden! Wegen der Nutzung vieler verschiedener Medien spricht man auch von Mixed Media.

Im Bereich der Publizistik ist unter Multimedia folgendes gemeint:

“Unternehmen oder Unternehmensgruppen, die mit mehreren, auch unterschiedl. Medienbetrieben ... tätig sind.” [Bro93]

Hier wird unter Multimedia die Kooperation von Unternehmen mit mehreren Medienbetrieben verstanden.

Für den schulischen Bereich findet sich folgende Definition:

“Im Unterricht spricht man von Medienverbund oder M.-System, wenn zur Erarbeitung eines Stoffes versch. Medien kombiniert werden.” [Bro93]

Als schultypische Medien sind dabei bspw. Dias, Audio- und Videokassetten gemeint.

Desweiteren findet man den Multimediabegriff in der Unterhaltungsbranche:

“...die aufeinander abgestimmte Verwendung versch. (vieler) Medien, Medienverbund, bes. in der Unterhaltungsbranche.” [Bro91, Bro93]

Der Multimediabegriff wird also im Kunst-, Publizistik-, Schul- und Unterhaltungsbereich unterschiedlich verwendet. Was unter Multimedia im informationstechnischen Sinne gemeint ist, wird im folgenden Abschnitt erläutert.

1.3.1.2 Multimedia - Begriffsbildung

Heute wird Multimedia zumeist im informationstechnischen Kontext gebraucht, was durch die informationstechnische Penetration des Menschen begründet werden kann.

Die zunehmende Penetration und einhergehende Bedeutung der Informationstechnik zeigt sich auch in Lexika: Die allgemeine Brockhaus-Enzyklopädie enthielt 1991 keine, speziell auf die Informationstechnik zugeschnittene, Multimediadefinition. Schon zwei Jahre später wurde die Begriffserklärung von Multimedia, wie folgt, erweitert:

“...elektron. Medium (z.B. CD-ROM), das Text, Bild, Ton, Video oder mindestens zwei dieser Elemente beinhaltet” [Bro93]

Diese Definition beinhaltet schon informationstechnische Fachwörter. Desweiteren werden die elektronischen Medien (CD-ROM) und die Darstellungsmittel für Daten (Text, Bild, Ton und Video) präzisiert.

Durch einen Blick in fachbezogene Lexika kann der Multimediabegriff noch weiter präzisiert werden. Das Lexikon für Computergrafik und Bildverarbeitung definiert ein Multimedia-System folgendermaßen:

“Auf Computertechnik basierendes System zur integrierten Verarbeitung und Darstellung von Informationen in mehreren Medien (Text, Grafik, Standbild, Computer Animation, Video, Audio), wobei diese nicht starr miteinander gekoppelt sind.” [Lex94]

Der Einsatz von Multimedia muß also rechnergestützt erfolgen, d.h. die Medien werden digital verarbeitet und dargestellt. Die Verarbeitung einzelner Medien erfolgt integriert, aber dennoch unabhängig voneinander. D.h. man kann mehrere Medien bündeln, wobei diese untereinander in keiner funktionalen Abhängigkeit stehen.

Im Kleinen Lexikon der Informatik findet man für Multimedia folgende Erklärung:

“Sammlung von verschiedenen Datentypen, Beziehungen zwischen verschiedenen Datentypen, wie Synchronisation, Kombination und Konvertierung von einem in das andere Medium.” [Kle95]

Hier ist unter 'media' eine Vielzahl von Datentypen zu verstehen die untereinander in einer Beziehung stehen. Sie müssen synchronisierbar, kombinierbar und konvertierbar sein. Unter Synchronität ist bspw. das korrekte zeitliche Einsetzen eines Musikstückes zu einem Video zu verstehen. Eine Synchronisierung kann aber auch räumlich oder inhaltlich erfolgen. Die oben angegebenen Datentypen werden gewöhnlich als Repräsentationsmedien bezeichnet, da sie die rechnerinterne Darstellung eines Mediums repräsentieren.

Zusammenfassend läßt sich folgendes festhalten: Eine allgemeingültige und -umfassende Definition von Multimedia kann nicht gegeben werden. Eine Erklärung hierfür ist, daß Multimedia ein relativ junges und zudem ein sich schnell veränderndes, wachsendes Teilgebiet der IT ist.

Die betrachteten Definitionen stellen nur einen Teilaspekt von Multimedia dar.

Folgende Forderungen an Multimedia kann trotzdem festgehalten werden: Texte, Bilder (bzw. Grafiken), Töne (Musik, Sprache und Geräusche) und Video sind eingesetzte Mittel um Daten zu repräsentieren. Diese (sog. Perzeptionsmedien) werden durch die Sinnesorgane des Menschen wahrgenommen (Abb.1.4). Die Medien sollen synchronisierbar sein, voneinander unabhängig und rechnergestützt integriert verwaltet werden können.

Dies veranlaßt zu folgender, auf [Ste95] basierter Definition:

Multimedia sind mehrere diskrete und kontinuierliche Perzeptionsmedien, die rechnergestützt und integriert verwaltet werden aber dennoch unabhängig voneinander agieren können.

Medien können unter drei verschiedenen Aspekten betrachtet werden: unter den Hardware-Aspekt, dem Aspekt der internen Repräsentation und dem Aspekt der Daten- bzw. Informationsdarstellung.

1.3.1.3 Virtual Reality

Von Multimedia abzugrenzen ist derzeit noch Virtual Reality. Obwohl eine Annäherung zu erkennen ist. Unter Virtual Reality (VR) versteht man eine künstlich geschaffene audiovisuelle Welt. Die darin enthaltenen Objekte sollen dabei interaktiv mit den Nutzer agieren. Diese fehlende Interaktivität grenzt Multimedia noch von VR ab.

Als Programmiersprache zur Visualisierung wird heute unter anderem VRML (Virtual Reality Modeling Language) eingesetzt, mit der auch eine auditive Untermalung möglich ist.

Neu an VR ist die Hinzunahme der dritten Dimension; womit sich neue Gestaltungsmöglichkeiten ergeben, da man bisher bei der visuellen Darstellung zweidimensional programmiert hat.

1.3.2 Ziele von Multimedia-Systemen

Knaurs Lexikon erklärt die Zielsetzung multimedialer Nutzung folgendermaßen:

“Multimedia, Mixed Media, Medienverbund, gemeinsame Benutzung verschiedener, sich ergänzender Medien zur Erreichung des gleichen Zwecks.” [Kna95]

Da hier auf den informationstechnischen Aspekt von Multimedia eingegangen wird, ist unter der allgemeinen Zielsetzung das Erreichen einer adequaten Informationsvermittlung und -darstellung zu verstehen.

So ist bspw. die Wettervorhersage multimedial: Meßdaten sind quantitativ gar nicht erfaßbar - sie werden durch Farbanimationen ersetzt. Die visuelle Darstellung der Daten kann besser durch den Menschen perzipiert werden.

Ein weiteres Ziel von Multimedia ist die Verbesserung der Mensch-Computer-Schnittstelle, die durch die multimedialen Mittel an neuen Gestaltungsmöglichkeiten gewinnt. In Zukunft sollen bspw. Computer allein durch Sprache und Gestik (Handbewegungen) gesteuert werden.

Multimedia kann auch zur Verbesserung der Mensch-zu-Mensch-Kommunikation genutzt werden. Durch Multimedia erfolgt eine verständlichere und damit sichere Informationsübermittlung. Abb. 1.2 illustriert, daß man allein durch Hören oder Sehen nicht zu einer gewünschten Informationsübermittlung bzw. Verständigung gelangt. Durch Kombination sich gegenseitig ergänzender Medien kann dieses Ziel viel besser erreicht werden. Zudem gewinnen die Daten bzw. Informationen beim Einsatz multimedialer Mittel an Klarheit und Anschaulichkeit da mehrere Sinnesorgane daran perzipieren können.



Abbildung 1.2: Ziel von Multimedia

1.4 Multimedia-Mittel, -Formate und -Hardware

In diesem Abschnitt werden die marktgängigen informationstechnischen Medien vorgestellt. Eine Unterteilung wird in Multimedia-Mittel, -Träger und -Formate vorgenommen.

Multimedia-Mittel sind Medien, die vom Empfänger direkt perzipiert werden. D.h. der Empfänger kennt keine Details über Datenträger (Online, Offline) oder Datenformate.

Multimedia-Formate beinhalten die von Sender zu übermittelten Daten. Ihre Notwendigkeit ergibt sich aus der Forderung, daß Multimedia-Mittel verwaltet werden müssen. Unter die Verwaltung fallen auch Aufgabenbereiche wie das Erstellen, Modifizieren und Speichern von Mitteln. Dies muß nach bestimmten Regeln, die durch die Formate gegeben werden, erfolgen. Ein Format kann dabei nur einem oder mehreren Mitteln zugeordnet sein. Die MPEG-Formate komprimieren bspw. Video- und Musikdaten.

Unter Multimedia-Hardware versteht man die Nutzung herkömmlicher Hardware für multimediale Zwecke.

Man erkennt also, daß die Übermittlung von Daten durch den Sender zum Empfänger in mehreren Zwischenschritten verläuft: Ein Bild wird bspw. ins PNG-Format transformiert und dann über ein Telekommunikationsnetz weitergeleitet. Beim Empfänger liegt es dann wieder im PNG-Format vor und wird wieder in ein Bild transformiert um wahrgenommen (perzipiert) werden zu können. Letzlich perzipiert der Empfänger somit die Multimedia-Mittel (Abb. 1.3).

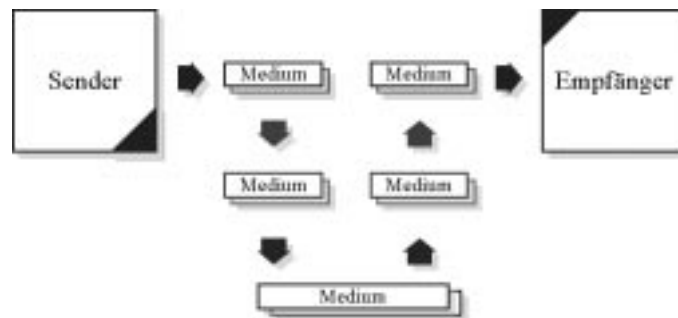


Abbildung 1.3: Datenübermittlung

Als Rezeptionsmittel besitzt der Menschen die Sinnesorgane. Die wichtigsten Sinnesorgane sind heute die Augen (zum Sehen) und Ohren (zum Hören). In Zukunft sollen auch die restlichen Sinnesorgane durch Multimedia-Mittel angesprochen werden können. Durch Datenhandschuhe sollen Hände fühlen und tasten und die Zunge durch Geschmacksplättchen schmecken können. Man hat lediglich keine Vorstellung wie man den Geruchssinn der Nase animieren könnte (Abb. 1.4).



Abbildung 1.4: Rezeptionsmittel

1.4.1 Multimedia-Mittel

Multimedia-Mittel sind zum einen in visuelle und auditive Medien und dazu orthogonal in diskrete (zeitunabhängige) und kontinuierliche (zeitabhängige) Medien unterteilbar (Abb. 1.5).

Im folgenden soll jedes Multimedia-Mittel kurz erläutert werden:

Text: Aneinanderreihung von Zeichen und Symbolen mit interpretierbarer Bedeutung.

	diskret	kontinuierlich
visuell	Text, Grafik, Bild	Laufschrift, Animationen, Video
auditiv		Musik, Sprache, Geräusche

Abbildung 1.5: Multimedia-Mittel

Bild, genauer: Digitalbild: Ein digitalisiertes Bild, das nicht auf dem Computer entstanden ist.

Grafik: Ein auf einem Computer entstandenes Bild.

Grafiken können in Pixel- oder Vektordarstellung vorliegen. Die Pixeldarstellung erfordert viel Speicherplatz, erlaubt aber fotorealistischere Bilder. Die Vektordarstellung beinhaltet mathematische Beschreibungen geometrische Objekte und spart dadurch Speicherplatz.

Animation: Sequenz von zusammengesetzten Einzelgrafiken.

Video: Abfolge von zusammengesetzten digitalisierten Einzelbildern.

Geräusche: Kurze, nicht unbedingt melodisch klingende auditive Folge.

Sprache: Kurze Folge von harmonisch klingenden Geräuschen.

Musik: Lange Folge von melodisch klingenden Tönen.

1.4.2 Datenformate und Kompressionsverfahren

Die meisten Datenformate verwenden eine Kompressionsverfahrens. Die Nutzung von Kompressionsverfahren ist durch die Notwendigkeit einer effizienten Nutzung von Systemressourcen (wie Rechenleistung, Speicherplatz oder Breitbandausnutzung) begründet.

Kompressionsverfahren kann man in verlustfreie (Entropiekodierung), verlustbehaftete (Quellkodierung) und hybride Verfahren (Kombination aus Quellen- und Entropiekodierung) unterteilen.

Entropiekodierungen beachten beim Komprimieren die Struktur der Daten. Dadurch ist eine spätere verlustfreie Rekonstruktion möglich.

Verlustbehaftete Verfahren komprimieren unter Beachtung der spezifischen Semantik der Daten und des Mediums. Der Kompressionsgewinn wird im wesentlichen durch Anpassung der Datenflüsse an die menschlichen Sinnesorgane erreicht. Bspw. werden Frequenz, die ober- oder unterhalb einer bestimmten Frequenzhöhe liegen vom menschlichen Ohr nicht wahrgenommen und deshalb verworfen.

Weitere Datenkompressionsverfahren sind die fraktale Kompression und Wavelets. Eine Überblick über Kompressionsverfahren wird in Abb. 1.6 gegeben.

Für Texte sind ASCII (American Standard Code for Information Interchange) und EBCDIC (Extended Binary Coded Dezimal Interchange Code) die vorherrschenden Formate. Wobei zunehmend ASCII gegenüber EBCDIC, der von IBM eingeführt wurde, favorisiert wird. Auf beide kann Entropiekodierung angewendet werden. Als neuer Textformat gilt der Unicode, der bspw. bei Java Verwendung findet.

Die verbreitesten Bild- und Grafikformate sind GIF und die JPEG-Formate. Zukünftig wird GIF (Graphic Interchange Format) jedoch wahrscheinlich von anderen Formaten verdrängt werden. Da die Firma CompuServe von ihrem Patentrecht auf GIF Gebrauch machen will und bei weiterer Verwendung eine monetäre Schutzgebühr von deren Nutzern erheben will. Ein möglicher Nachfolger ist das PNG-Format (Portable Network Graphics), auf das später genauer eingegangen wird.

JPEG steht für 'Joint Photographic Experts Group'. Die JPEG-Formate gehören zu den hybriden Kompressionsverfahren. Sie werden vorwiegend für die Bildkompression eingesetzt, wobei mit MJPEG, einer

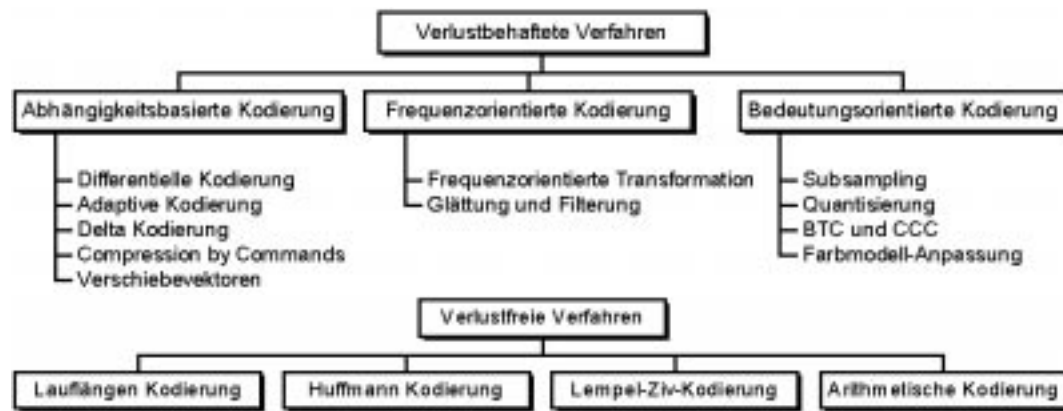


Abbildung 1.6: Kompressionsverfahren

Variante von JPEG, auch Animationen komprimiert werden.

Die verbreitesten Videoformate sind die MPEG-Formate (Motion Picture Experts Group), H.261 bzw. p*64 (von CCITT) und DVI (Digital Video Interactive von IBM/Intel). Alle drei gehören zu den hybriden Kompressionsverfahren.

Bei den Audioformaten sind Audio-Dateien (.au) bei UNIX und Wave-Dateien (.wav) in Windows/PC-Bereich vorherrschend. Diese Formate unterstützen alle auditiven Multimedia-Mittel (Geräusche, Sprache und Musik).

Die bekanntesten 'Multimedia-Sprachen' sind HTML, Java und VRML. HTML (Hypertext Markup Language) ist eine Strukturbeschreibungssprache und erlaubt kleine Multimediaprogramme (Text, Audio und Animationen mit GIF89a). Desweiteren können Java-Applets in HTML eingebunden werden.

Java ist eine von Sun entwickelte objektorientierte Programmiersprache. Sie ist plattformunabhängig, multithreaded, robust und sicher. Die mitgelieferten Bibliothek bietet Methoden mit der ein leichter Umgang mit Netzwerken, Fenstersystemen und Applets ermöglicht wird. Der Vorteil von Java ist die zur Verfügung stehende volle Funktionalität einer Programmiersprache.

1.4.3 Multimedia-Hardware

Bei Kauf eines sog. Multimedia-PCs (<http://www.spa.org/mpc/stand.htm>) erhält man neben dem Rechner samt Monitor eine Soundkarte, Aktivboxen und ein CD-ROM-Laufwerk. Damit sind auditive und visuelle Medien durch den Anwender perzipierbar. Mit dem CD-ROM-Laufwerk steht auch ein Speichermedium zur Verfügung, das den enormen Speicheraufwand multimedialer Anwendungen Rechnung trägt. Zunehmend werden auch Modems und Mikrofone zu Multimedia-PCs angeboten. Ein Händler versteht unter Multimedia also mehrere Hardware-Medien. In naher Zukunft wird auch Hardware, wie sie im VR-Bereich eingesetzt wird, angeboten. Daran erkennt man sehr gut, wie schnelllebig der Multimedia-Bereich ist.

Die eingesetzte Hardware ist folgendermassen klassifizierbar: Computer, Spezial-Hardware, Eingabe- und Ausgabegeräte, Speichermedien, Telekommunikationsnetze.

Die interessierende Kenngröße von Computern ist die Rechenleistung. Sie wird hauptsächlich von dem Prozessor bzw. den Prozessoren bestimmt, kann aber durch Spezial-Hardware erhöht werden. Die Spezial-Hardware setzt sich aus Steckkarten und Spezialprozessoren zusammen, die hauptsächlich zur Kompression und Digitalisierung eingesetzt werden (z.B. MPEG-De/Encoder).

In scheinbar unendlich vielen Variationen und Kombinationen scheinen E/A-Geräte vorhanden zu sein. Tabelle 1.7 zeigt eine grobe Auflistung heutiger und zukünftiger E/A-Geräte nach Multimedia-Mitteln geordnet an.

Multimedia-Mittel	Eingabe	Ausgabe
Text	Tastatur, Scanner (OCR)	Bildschirm, Drucker
Bild	Scanner, Kamera	Bildschirm, Drucker
Grafik	Maus-Zeiger, 3D-Zeiger	Bildschirm, Plotter, stereoskopische Brille für 3D-Effekt
Video	Video-Kamera	Bildschirm, Datenhelm
Audio	Mikrofon, Sound-Karte	Lautsprecher, Kopfhörer, Sound-Karte
Virtual Reality	Datenhandschuh	Holographie, Datenhelm

Abbildung 1.7: E/A-Geräte

Zukünftig werden sprachliche und sensorische Ein- und Ausgabemittel zunehmen. Bspw. forscht man momentan an sog. Geschmacksplättchen, die die Geschmacksnerven der Zunge stimulieren sollen.

Einen wichtigen Bereich bilden die Massenspeicher. Auf Primärspeicher (Disketten) und Archivspeicher (DAT, Bänder) wird nicht näher eingegangen. Primärspeicher besitzen für den multimedialen Einsatz zu wenig Speicherkapazität und eine unzureichende Datenübertragungsleistung. Archivspeicher bieten nur einen sequentiellen Speicherzugriff.

Heutige, im Einsatz befindliche Massenspeicher sind magnetische und laseroptische Plattenlaufwerke. Magnetplatten kann man in Festplatten und Wechselplatten unterteilen. Die magnetisierbaren Kreisscheiben von Festplatten sind stets verankert, während sie bei Wechselplatten gewechselt werden können. Zu den Wechselplatten zählen SyQuest-, Bernoulli- und die Jaz-Drives. Laseroptischen Platten haben sich erst in den letzten Jahren als Speichermedien etabliert. Die bekanntesten Vertreter sind CD-ROMs (Compact Disk, ca. 650 MByte) und deren Abkömmlinge. Sie liegen, was die Zugriffsgeschwindigkeit und Speicherkapazität betrifft, hinter den Magnetplatten zurück. Ihre mittlerweile enorme Verbreitung läßt sich durch relativ niedrige Beschaffungskosten erklären. Die MOs (Magnet Opticals) basieren auf einem hybriden Verfahren, d.h. beim Schreiben wird die CD erhitzt und magnetisiert. Für Anfang 1997 ist die CD-RW (Re-Writeable) angekündigt. Sie soll mehrmals beschreib- und lesbar sein und ca. 680 MByte speichern können [Com].

Speichermedien bilden heute bzgl. der Übertragungsgeschwindigkeit einen Leistungsengpaß. Benötigte Datenmengen um bspw. ein Video in Echtzeit abzuspielen, können nicht schnell genug geladen werden. Aus diesem Grund, und um Speicherplatz zu sparen, werden Kompressionsverfahren eingesetzt. Daten werden komprimiert abgelegt und geladen und brauchen im Bedarfsfall nur noch dekomprimiert werden, was in Echtzeit möglich ist.

Häufig sind nicht alle benötigten Daten (bspw. bei verteilten Systemen) auf dem lokalen Speichermedium vorhanden. Über Telekommunikationsnetze können die Daten geholt werden. Da die Übertragungsgeschwindigkeit von Netzen geringer ist als die von Speichermedien, ergibt sich zwangsläufig auch ein Engpaß. Große Datenmengen versucht man über sog. Hochgeschwindigkeitsnetze zu übertragen. Einen Überblick über Hochgeschwindigkeitsnetze gibt folgende Tabelle 1.8:

Bemerkung: Heutige Anforderungen, die multimediale Anwendungen bzgl. Rechen- und Übertragungsleistung im Hardware- und Netzwerkbereich stellen, sind erfüllbar. An der Software scheitert aber deren Nutzung.

Netzwerktechnik	Beschreibung
(B)-ISDN (Integrated Services Digital Network)	als Ersatz für bisheriges Telefonnetz gedacht, als Breitband-ISDN für Backbone ausgelegt. 64 KBit/s bzw. 622 MBit/s
FDDI (Fibre Distributed Data Interface)	ausgereifte Technologie, hervorragend für Backbone geeignet. 100 MBit/s
ATM (Asynchronous Transfer Mode)	universelle Netzwerkarchitektur für Übertragung, Basis für multimediale Anwendungen (Videokonferenzen), technisch nicht ganz ausgereift und nicht vollständig standardisiert. bis zu 622 MBit/s
Fast-Ethernet und Gigabit-Ethernet	kostengünstig, leichte Migration, Standard für Anfang 1998 erwartet. 100 bzw. 1000 MBit/s

Abbildung 1.8: Hochgeschwindigkeitsnetze

1.5 Entwicklung multimedialer Anwendungen

Multimedia-Software ist in Anwendungsprogramme und Entwicklungsprogramme klassifizierbar.

Die Anwendungsprogramme werden nach gängigen Softwareentwicklungsmodellen (Wasserfallmodell, Transformationsmodell, Risikomodell) erstellt. Dabei sind folgende Probleme (besonders bei großem Multimediaeinsatz) zu beachten: Die Prozessorleistung und die interne Datenübertragung im Computer sind unzureichend. Zumeist reicht auch die Kapazität des lokalen Speichermediums nicht aus. Durch die Nutzung von Rechnernetzen (bspw. bei einer Fragmentierung der Daten) erhöht sich, bedingt durch die anfallende Kommunikationszeit, der Leistungsengpaß. Desweiteren sind keine leistungsstarken Datenbanksysteme für das Verwalten (Ablegen, Wiederauffinden und Aktualisieren) großer Datenbestände vorhanden. Ein softwaretechnisches Problem ergibt sich durch die notwendige Synchronisierung von Medien. Die Synchronisierung trägt einen erheblichen Teil zur Komplexität von Multimediaanwendungen bei, da man zeitliche, räumliche und inhaltliche Synchronisierungspunkte setzen muß.

Hilfsmittel, mit denen Multimediaanwendungen leichter erstellt werden können, bezeichnet man als Autorensysteme (Authoring Systems, Authoring Tools). Man kann sie als Entwicklungswerkzeuge betrachten, die speziell auf die Problematik von Multimediaanwendungen eingehen.

Autorensysteme unterscheidet man folgendermassen:

Seitenorientierte Systeme: Die Grundkonstrukte dieser Systeme werden als Seiten eines Buches, zwischen denen Verbindungen bestehen, betrachtet. In den Seiten werden Komponenten, wie bspw. Textblöcke und Grafiken plziert.

Flußdiagrammorientierte Systeme: Hier werden zu einem Flußdiagramm Piktogramme hinzugefügt. Das Flußdiagramm bestimmt den Ablauf des multimedialen Systems. Die Piktogramme stellen einzelne Funktionen dar, die vom System angeboten werden.

Zeitbasierte bzw. Präsentationssysteme: Der Ablauf des System wird durch eine Zeitachse dargestellt. An dieser Zeitachse werden die Elemente des Systems angeordnet. Durch Start- und Endpunkte einer Sequenz erfolgt die Synchronisation.

Bemerkung: Ein Problem, daß alle Multimediadokumente gemein haben, ist die korrekte Umsetzung der einzelnen Medien auf Papier. Auf Hypertext basierte Dokumente sind wegen der Links nicht ohne weiteres umsetzbar, da ein Teil der Funktionalität verloren wird. Audio- und Video-Daten sind auf

Papier gar nicht umsetzbar.

1.6 Nutzung von Multimedia und Electronic Publishing

Der Einsatz multimedialer Systeme, d.h. Systeme die sich multimedialer Techniken bedienen, ist in folgende drei Bereiche unterteilbar:

Lehre, Ausbildung und Training (CBT, Computer Based Training),
Präsentation von Produkten, Dienstleistungen, Firmen, Institutionen und
Präsentation und Dokumentation von Objekten, Prozessen, Sachverhalten.

Die finanziell rentable Nutzung von Multimedia ist in CBT-Bereich anzusiedeln. Der heutige europäische CBT-Markt ist in Abb 1.9 skizziert:



Abbildung 1.9: CBT - Europäischer Markt 1996

Nach der Studie "Multimedia in Training and Education" des englischen Marktforschungsinstitut Data-monitor soll Deutschland zum größten CBT-Markt bis zum Jahr 2005 avancieren. Der Gesamtumsatz soll bei 11,5 Mrd Mark liegen.

Heute nutzen schon viele Unternehmen Multimedia bzw. Multimedia-Hardware. Gemäß der Studie "Multimedia-Barometer" von GfK, MGM und Horizont liefern deutsche Medien- und PR-Unternehmen das in Abb. 1.10 Nutzungsprofil:

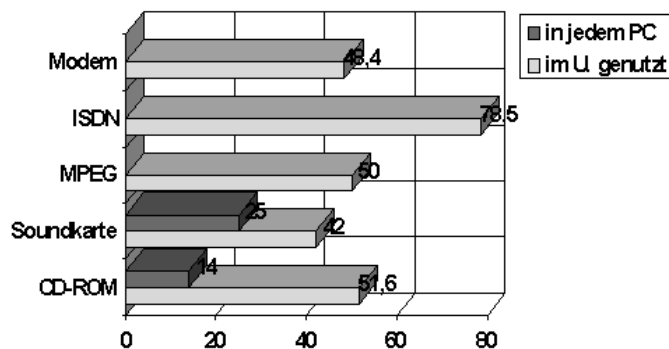


Abbildung 1.10: Marktzahlen

1.7 Ausblick

Der Einsatz von Multimedia wird dabei als defensives marketingpolitisches Instrument eingesetzt, d.h. man versucht den Kunden an das Unternehmen bzw. Produkt durch Kundenbetreuung zu binden. Achtzig Prozent der Unternehmen wollen die Budgets sogar noch erhöhen, davon 33% um bis zu 50%. Insgesamt nutzen 93% aller Unternehmen Multimedia.

In den nächsten zehn Jahren sollen allein in den EU-Mitgliedstaaten rund eine Millionen zusätzliche Arbeitsplätze in EP-Bereich entstehen. Zu diesem Ergebnis kommt eine Studie, die Anderson Consulting mit IENM (Institut für Informationswirtschaft und neue Medien) für die Europäische Kommission durchgeführt hat.

In Zukunft wird die steigende Rechenleistung, Speicherkapazität sowie Kommunikationsgeschwindigkeit die multimediale Entwicklung weiter vorantreiben.

Die Offline-Medien werden zunehmend an Bedeutung verlieren. Die CD-ROM wird in den kommenden zehn bis fünfzehn Jahren durch neue Speichertechnologien ersetzt werden.

Weitere Spezial-Hardware für Multimedia wird entwickelt werden, wie bspw. der MMX von Intel (<http://www.intel.com>).

Der PC- und TV-Bereich werden ineinander verschmelzen. Booz, Allen & Hamilton sagen voraus, daß im Jahr 2000 allein in Deutschland rund sechs Milliarden Mark mit Pay Per View und Video On Demand umgesetzt werden. Zudem ist ein allgemeiner Trend zu Service On Demand erkennbar.

Immer mehr Bereiche des täglichen Lebens werden von Multimedia beinflußt werden.

1.8 Literatur

[Bro91] *Brockhaus-Enzyklopädie in 24 Bnd.* Brockhaus Verlag, 1991.

[Bro93] *Brockhaus-Enzyklopädie in 24 Bnd.* Brockhaus Verlag, 1993.

[Com] *Computerwoche.*

[Hol95] Wieland Holfelder. *Multimediale Kiosksysteme.* Vieweg, Braunschweig/Wiesbaden, 1995.

[Kna95] *Knaurs Lexikon.* 1995.

[Lex94] *Lexikon fuer Computergrafik und Bildverarbeitung.* Vieweg-Verlag, 1994.

[Lex95] *Kleines Lexikon der Informatik.* Oldenburg-Verlag, 1995.

[Ste95] Steinmetz. *Multimedia.* Addison Wesley Longman, 1995.



Kapitel 2

PNG

Matthias Baas

2.1 Übersicht

In diesem Kapitel wird das Dateiformat PNG vorgestellt, das als Nachfolger für GIF entwickelt wurde. Es dient der verlustlosen Speicherung von Einzelbildern und soll besonders in Netzwerk-Anwendungen, wie z.B. dem World-Wide Web, zum Standard-Grafikformat werden.

Die folgenden Abschnitte erläutern den allgemeinen Aufbau einer PNG-Datei und die Möglichkeiten, die das Format bietet. Es wird dabei auf die Kompression, das Interlacing, die Transparenz und die Möglichkeiten bei der Farbwiedergabe eingegangen.

2.2 Einleitung

Was ist PNG?

PNG (gesprochen: „ping“) steht für *Portable Network Graphics* und ist ein neues Dateiformat für Bilddaten. Es wurde Anfang 1995 entwickelt und soll das Graphics Interchange File (GIF) Format ersetzen, das bisher als Standardformat für verlustlose Bildspeicherung galt.

Warum PNG?

Es gibt bereits eine Vielzahl von Grafikformaten mit denen Bilder auf unterschiedlichste Weise gespeichert werden können. Warum hat man sich also dennoch entschieden, ein komplett neues Grafikformat zu entwickeln?

Der Auslöser dafür kam von CompuServe, dem Urheber von GIF. Ende Dezember 1994 erklärte CompuServe, daß für das bisher frei verfügbare GIF-Format von nun an neue Lizenzrechte gelten. Hersteller von Programmen, die das GIF-Format verwenden wollen, haben also dafür ein Entgelt an CompuServe zu zahlen. Diese Änderung empörte natürlich viele Software-Entwickler, die das GIF-Format seit seiner Entwicklung 1987 in ihren Programmen frei verwendet haben. Selbst CD-ROM Hersteller, die Public Domain Programme mit GIF-Unterstützung veröffentlichen, müssen jetzt für diese Programme bezahlen. Dies alles wurde auch in dem Usenet Diskussionsforum **comp.graphics** besprochen, was dazu führte, daß man sich zusammentat, um einen Weg zu suchen, das GIF-Format zu ersetzen.

Zunächst versuchte man, sich auf ein bereits bestehendes Dateiformat zu verlagern, erkannte jedoch schnell, daß keines der Formate all die Dinge enthielt, die GIF so populär gemacht haben. Allem voran sollte das Format portabel sein. Formate wie z.B. BMP, IFF, PCX, PICT sind deshalb nicht geeignet, da sie für ein bestimmtes Betriebssystem geschaffen worden sind, wobei die Austauschfähigkeit nicht im Vordergrund stand. Bleiben also nur noch die Formate, die Bilder in einer vom Betriebssystem unabhängige Weise speichern, wie z.B. das Tagged Image File Format (TIFF) von Microsoft/Aldus. TIFF ist allerdings so komplex, daß nahezu kein Programm alle möglichen TIFF Optionen unterstützt. Es kann daher leicht passieren, daß ein TIFF-Bild, das von Programm A erstellt wurde, von Programm B nicht gelesen werden kann. Ein weiterer Nachteil von TIFF ist, daß man ein Bild nicht sequentiell einlesen kann. In Netzwerk-Anwendungen wie z.B. WWW-Browsern heißt das, daß die Datei erst komplett übertragen werden muß, bevor das Bild dargestellt werden kann. TIFF wäre also auch kein

passender Ersatz für GIF. Als nächstes betrachtete man das JPEG File Interchange Format (JFIF), das als sequentielles Format entwickelt wurde und eine sehr hohe Kompressionsrate erzielt. Allerdings ist diese Kompression nicht verlustlos, was vor allem bei Zeichnungen, Liniengrafiken, ... zu einer ziemlich schlechten Bildqualität führen kann. Es existiert zwar die Option, Bilder verlustlos zu speichern, jedoch wird diese Option nur von sehr wenigen Programmen unterstützt.

Also kam man zu dem Schluß, daß ein neues Format kreiert werden muß, und zwar eines, das die folgenden Eigenschaften hat:

- Einfach, sauber und leicht zu implementieren.
- Portabel.
- Frei verfügbare Quelltexte zum Lesen und Schreiben von Bildern.
- Existierende GIF-Bilder müssen zu 100% konvertiert werden können.
- Sequentielles Lesen und Schreiben muß unterstützt werden.
- Bessere Kompression als GIF, wobei immer noch verlustlos komprimiert werden soll.
- Inkrementelles Anzeigen (progressive display) und Transparenz sollte mindestens so wie im GIF-Format möglich sein.

Als Ergebnis kam das PNG-Format heraus, mit dem diese und weitere Ziele erreicht wurden. Das Wichtigste ist hierbei, daß das Format frei von Rechten Dritter ist, es kann daher in Programmen ohne rechtliche Probleme implementiert werden.

Was bietet PNG?

Hier sind nun stichwortartig einige der Punkte aufgelistet, die in der bisherigen PNG-Spezifikation enthalten sind. Das Format kann aber auch leicht erweitert werden, ohne dabei die Austauschfähigkeit zu verlieren.

GIF-Eigenschaften, die auch in PNG enthalten sind:

- Bilder mit Farbindizes bis zu 256 Farben.
- Serialisierbarkeit (streamability): Dateien können seriell gelesen und geschrieben werden.
- Inkrementelles Anzeigen: Eine entsprechend vorbereitete Datei kann während der Übertragung dargestellt werden, so daß man sehr schnell das Bild in grober Auflösung hat, welches dann schrittweise verfeinert wird.
- Transparenz: Teile des Bildes können als transparent markiert werden, um z.B. ein nicht-rechteckiges Bild zu erzeugen.
- Zusatzinformation: Textuelle Kommentare und andere Daten können in der Bilddatei gespeichert werden.
- Komplett unabhängig von Hardware und Betriebssystem.
- Effektive, 100% verlustlose Kompression.

Einige neue Eigenschaften von PNG, die in GIF nicht enthalten sind:

- Echtfarbenbilder mit bis zu 48 Bit pro Bildpunkt.
- Graustufenbilder mit bis zu 16 Bit pro Bildpunkt.
- Kompletter Alpha-Kanal (allgemeine Transparenz-Maske).

- Gamma-Information wird mitgespeichert, damit läßt sich die Helligkeit und der Kontrast so anpassen, daß das Bild auf verschiedenen Systemen in gleicher Weise angezeigt wird.
- Zuverlässiges Erkennen beschädigter Dateien (z.B. durch Übertragungsfehler).
- Schnellere Anzeige der ersten groben Auflösung bei der inkrementellen Anzeige als bei GIF.

2.3 Aufbau einer PNG-Datei

Die in einer PNG-Datei gespeicherten Daten sind in Blöcke unterschiedlicher Länge gegliedert. Jeder Block enthält eine spezielle Art von Information, wie z.B. die eigentlichen Bildpunkte, textuelle Kommentare, Farbtabelle, usw.

Jede PNG-Datei besteht aus einer *Signatur* und einer Folge beliebig vieler Blöcke, den *Chunks*. Die Signatur dient lediglich dazu, eine PNG-Datei als solche zu identifizieren, während in den Chunks die eigentlichen Bildinformationen gespeichert werden.

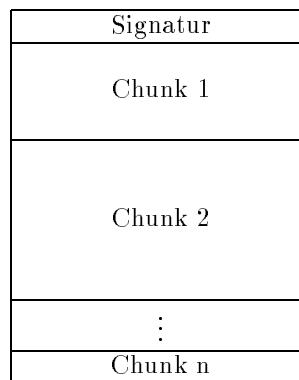


Abbildung 2.1: Allgemeine Dateistruktur

2.3.1 Signatur

Die ersten 8 Byte einer PNG-Datei bilden die Signatur, die sich aus den in Abbildung 2.2 stehenden Werten zusammensetzt.

Dezimal	137	80	78	71	13	10	26	10
Hexadezimal	89	50	4e	47	0d	0a	1a	0a
ASCII C Notation	\211	P	N	G	\r	\n	\032	\n

Abbildung 2.2: Signatur einer PNG-Datei

Diese 8 Byte identifizieren eine PNG-Datei und dienen gleichzeitig dazu, Übertragungsfehler zu entdecken. Bei der Wahl der Signatur wurden folgende Überlegungen zugrunde gelegt:

- Das erste Byte sollte ein nicht-ASCII Zeichen sein, damit reine Text-Dateien nicht fälschlicherweise als PNG-Datei identifiziert werden können. Zusätzlich werden durch das erste Byte Fehler aufgefangen, die dadurch entstehen, daß ein Übertragungsverfahren Bit 7 löscht.
- Die Bytes 2 bis 4 benennen das Format.

- Byte 5 und 6 bestehen aus einer CR-LF (Carriage Return - Line Feed) Kombination, die unter MS-DOS das Ende einer Textzeile kennzeichnet. Wird nun eine PNG-Datei fälschlicherweise als Textdatei übertragen, kann es vorkommen, daß solche CR-LF Sequenzen in ein einzelnes LF umgewandelt werden, was zu fehlerhaften Bilddateien führen würde.
- Byte 7 besteht aus dem Zeichen Control-Z, was beim Betrachten der Datei unter MS-DOS dazu führen würde, daß die Betrachtung abgebrochen wird.
- Byte 8 ist schließlich ein einzelnes LF, was im Prinzip die gleiche Aufgabe hat wie Byte 5 und 6. Es soll also frühzeitig erkannt werden, ob einzelne LF in CR-LF Sequenzen umgewandelt werden.

2.3.2 Chunks

Chunks enthalten sämtliche Informationen, die in einer PNG-Datei gespeichert werden können. Dabei werden unterschiedliche Arten von Informationen auch in unterschiedlichen Chunks mit unterschiedlichen Typen gespeichert. Jeder Chunk besteht aus den vier in Abbildung 2.3 gezeigten Komponenten.

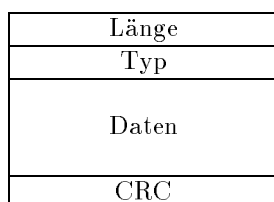


Abbildung 2.3: Aufbau eines Chunks

2.3.2.1 Länge

Die erste Komponente ist eine 4 Byte Integer-Zahl, die die Länge des Datenteils enthält. Dabei zählen *nur* die Daten selbst, nicht die Länge des gesamten Chunks.

2.3.2.2 Typ

Als Typkennung wird eine 4 Byte Integer-Zahl verwendet, die festlegt, wie die im Chunk enthaltenen Daten zu interpretieren sind. Die einzelnen Bytes dürfen jeweils nur aus den ASCII-Zeichen a-z bzw. A-Z bestehen, zusammen bilden sie also den vier Zeichen langen Namen des Chunks.

Der Chunk-Name unterliegt einigen Regeln, die es einem Programm beim Einlesen eines Bildes erlauben, auch unbekannte Chunks handzuhaben. Die Regeln beziehen sich auf die Groß-/Kleinschreibung der einzelnen Buchstaben, oder anders gesagt, auf Bit 5 der einzelnen Bytes. Im ASCII-Zeichensatz unterscheiden sich die Groß- von den Kleinbuchstaben nur im 5. Bit, ist es gesetzt handelt es sich um einen Kleinbuchstaben, andernfalls um einen Großbuchstaben. Damit können also in dem vier Zeichen langen Chunk-Namen vier Bits kodiert werden, die allein durch die Groß-/Kleinschreibung abgelesen werden können. Abbildung 2.4 zeigt, welche Bits in welchem Zeichen kodiert werden.



Abbildung 2.4: Beispiel für Namensgebung anhand des tEXt-Chunks

Verzichtbarkeits Bit (ancillary bit): Ist dieses Bit gesetzt, so handelt es sich um einen Zusatzchunk (siehe Abschnitt 2.3.4), andernfalls um einen unverzichtbaren Chunk (siehe Abschnitt 2.3.3). Unbekannte Zusatzchunks können beim Einlesen ignoriert werden, unverzichtbare Chunks enthalten dagegen Informationen, die zum korrekten Darstellen eines Bildes benötigt werden.

Privat Bit (private bit): Bei Chunks, die in der offiziellen PNG-Spezifikation enthalten sind, ist dieses Bit gelöscht. Anwendungen, die eine PNG-Datei um eigene Chunks erweitern, müssen dieses Bit setzen. Dadurch wird gewährleistet, daß es zu keinen Namenskonflikten zwischen offiziellen und privaten Chunks kommt.

Reserviertes Bit (reserved bit): Dieses Bit ist für zukünftige Erweiterungen reserviert und muß immer gelöscht sein, es hat derzeit noch keine Bedeutung.

Kopier Bit (safe-to-copy bit): Dieses Bit regelt, ob unbekannte Chunks bei Modifikation des Bildes kopiert werden dürfen oder nicht. Falls ein Programm einen *unverzichtbaren* Chunk auf irgendeine Weise ändert, dürfen unbekannte Chunks nur kopiert werden, wenn ihr Kopier Bit gesetzt ist. Andernfalls hängen die Daten des unbekannten Chunks von den unverzichtbaren Chunks ab. Bei unverzichtbaren Chunks ist dieses Bit immer gelöscht.

2.3.2.3 Daten

In diesem Teil des Chunks stehen die eigentlichen Daten. Wie die Daten zu interpretieren sind, hängt vom Typ des Chunks ab.

2.3.2.4 CRC

Am Ende eines Chunks ist ein 4 Byte CRC (Cyclic Redundancy Check) gespeichert, der nach den standardisierten CRC Methoden berechnet wird (siehe [IS 84] oder [Int94]). Mit diesem Wert läßt sich prüfen, ob ein Chunk Fehler enthält, z.B. Fehler durch falsche Übertragung. Außerdem kann erkannt werden, ob ein Programm einen Chunk verändert hat, indem der neue und alte CRC-Wert verglichen wird. Hat sich der Wert geändert, so wurde auch der Chunk verändert.

Der Test bezieht sich auf den Chunk-Typ und die Daten, jedoch nicht auf die Chunk-Länge. Dies ist allerdings keine „Sicherheitslücke“, da bei falscher Länge ja auch nicht auf den richtigen CRC-Wert zugegriffen wird und somit der Fehler erkannt wird.

2.3.3 Unverzichtbare Chunks

Kommen wir nun zu der Speicherung der eigentlichen Bildinformation. Sämtliche Information, die zum korrekten Darstellen eines Bildes unbedingt erforderlich ist, steht in den *unverzichtbaren* Chunks (critical chunks). In jeder PNG-Datei müssen mindestens diese Chunks enthalten sein und jedes Programm, das mit PNG-Dateien arbeitet, muß sie kennen und auswerten können.

2.3.3.1 IHDR - Image Header

Der IHDR-Chunk bildet den Kopf der Datei, er *muß* als erster Chunk direkt nach der Signatur stehen und enthält die in Tabelle 2.1 aufgeführten Informationen.

Die Breite und Höhe wird jeweils in Bildpunkten angegeben. Die Bittiefe gibt die Anzahl Bits pro Farbwert an, dies ist nicht zwingenderweise identisch mit der Anzahl Bits pro Bildpunkt. Welche 2er-Potenzen bei der Bittiefe wirklich gültig sind, hängt von dem verwendeten Farbtyp ab (Tabelle 2.2).

Der Farbtyp legt fest, wie die einzelnen Bildpunkte abgespeichert sind und wie ihre Werte zu interpretieren sind. Er ist eine Kombination der drei Werte 1 (es wird eine Palette benutzt), 2 (das Bild enthält

Länge	Name	Mögliche Werte
4 Byte	Breite	$1 - (2^{31} - 1)$
4 Byte	Höhe	$1 - (2^{31} - 1)$
1 Byte	Bittiefe	1,2,4,8,16
1 Byte	Farbtyp	0,2,3,4,6
1 Byte	Kompressionsmethode	0
1 Byte	Filtermethode	0
1 Byte	Interlacemethode	0,1

Tabelle 2.1: Header-Chunk

Farbtyp	Gültige Bittiefen	Interpretation
0	1,2,4,8,16	Jeder Bildpunkt enthält einen Grauwert.
2	8,16	Jeder Punkt enthält RGB-Werte.
3	1,2,4,8	Jeder Punkt enthält einen Palette-Index. Bei diesem Typ muß ein PLTE-Chunk vorhanden sein.
4	8,16	Jeder Punkt enthält einen Grauwert, gefolgt von einem Alpha-Wert.
6	8,16	Jeder Punkt enthält RGB-Werte, gefolgt von einem Alpha-Wert.

Tabelle 2.2: Farbtypen

„Farben“, im Gegensatz zu Grauwerten) und 4 (es wird ein Alpha-Kanal benutzt, siehe Abschnitt 2.6). Dadurch sind für den Farbtyp die in Tabelle 2.2 aufgelisteten Werte zulässig.

Die Werte 1, 5 und 7 sind nicht zulässig. Ein Wert von 1 oder 5 würde ja bedeuten, daß eine Palette ohne Farben benutzt werden soll, d.h. die Palette darf *nur* Grauwerte enthalten. Dies erreicht man aber auch, indem man in einer „farbigen“ Palette nur Grauwerte einträgt oder gleich ein Grauwert-Bild speichert. Der Wert 7 (Farbtabelle + Alpha Kanal) klingt im ersten Moment sinnvoll, man könnte ja auch in einem Bild des Typs 3 bestimmte Bereiche transparent machen wollen. Jedoch wurde dies von den Entwicklern des PNG-Formats auf andere Weise gelöst. Es gibt die Möglichkeit, neben der Farbtabelle noch eine „Alpha-Tabelle“ abzuspeichern, in der für jeden Farbindex ein Alpha-Wert gespeichert werden kann. Damit hat dann jeder Index eine eindeutige Farbe und ebenso einen eindeutigen Alpha-Wert (der genaue Aufbau des tRNS-Chunks befindet sich in [Gro96a]).

Das Feld Kompressionsmethode darf bisher nur den Wert 0 enthalten, d.h. es gibt nur eine Kompressionsmethode, mit der jedes Bild komprimiert sein muß. Andere Werte sind für zukünftige Erweiterungen reserviert.

Auch bei der Filtermethode ist bisher nur der Wert 0 erlaubt. Er steht für „adaptives Filtern mit 5 Grundfilter“, die bisher einzige Filtermethode. Ein Filter ist eine Methode, mit der die einzelnen Bildzeilen vor dem Komprimieren bearbeitet werden, um eine bessere Kompression zu erzielen (siehe Abschnitt 2.4.2).

Bei der Interlacemethode können zwei Werte angegeben werden: 0 (kein Interlace) und 1 (Adam7 Interlace). Auf das Interlacing wird in Abschnitt 2.5 näher eingegangen.

2.3.3.2 PLTE - Palette

Der Palette-Chunk liefert die Farbtabelle, die beim Farbtyp 3 unbedingt erforderlich ist und die in den Modi 2 und 6 (Echtfarben und Echtfarben + Alpha) optional verwendet werden kann. Er ist also nur bei

farbindizierten Bildern wirklich ein „unverzichtbarer“ Chunk, dennoch muß er von allen Programmen verstanden werden, da ansonsten manche PNG-Dateien nicht eingelesen werden könnten.

Die Tabelle enthält bis zu 256 Farbeinträge, wobei jede Farbe als RGB-Wert gespeichert wird (Abbildung 2.5).

R_0	G_0	B_0
R_1	G_1	B_1
R_2	G_2	B_2
R_3	G_3	B_3
\vdots	\vdots	\vdots

Abbildung 2.5: Palette-Chunk

Die RGB-Werte belegen *immer* jeweils 1 Byte, unabhängig von der verwendeten Bittiefe. Die Anzahl der Farben in der Palette wird anhand der Chunk-Länge ermittelt, wobei diese durch 3 teilbar sein muß. Ist ein Palette-Chunk in einer PNG-Datei vorhanden, so muß dieser dem ersten IDAT-Chunk vorausgehen. Mehrere Palette-Chunks sind nicht erlaubt.

Für den Farbtyp 3 gilt: Die erste Farbe in der Tabelle wird mit dem Index 0 angesprochen, die zweite mit 1, ... usw. Die Palette darf nicht mehr als 2^{Bittiefe} Einträge enthalten, kann jedoch durchaus weniger Farben umfassen. Im letzteren Fall gilt eine Referenz auf einen nicht vorhandenen Farbeintrag als Fehler.

Ein Palette-Chunk darf auch bei Echtfarbenbildern (Modus 4 und 6) mitgespeichert werden. Ist dies der Fall, wird durch die Tabelle ein Satz von Farben vorgeschlagen, mit dem die Farbanzahl des Bildes reduziert werden kann. Jede Farbe im Bild wird dabei auf denjenigen Farbeintrag in der Tabelle abgebildet, der der ursprünglichen Farbe am ähnlichsten ist. Das Bild kann dann auch auf Systemen dargestellt werden, die keine direkte Anzeige von Echtfarbenbildern erlauben.

2.3.3.3 IDAT - Image Data

IDAT-Chunks enthalten die Bilddaten. Wie die Daten hier gespeichert sind, hängt von den Einträgen im IHDR-Chunk ab.

Größe und Anzahl der IDAT-Chunks in einer Datei ist beliebig, sie müssen nur direkt aufeinander folgen. Es gibt auch keinerlei Beschränkung, an welcher Stelle die Daten getrennt werden dürfen, hier ist alles erlaubt. Zum Decodieren gilt also die Konkatenation der Daten aller IDAT-Chunks als Eingabestrom für den weiteren Bearbeitungsschritt.

2.3.3.4 IEND - Image Trailer

Dieser Chunk wird als letzter Chunk in einer PNG-Datei gespeichert, er markiert das Ende der Datei und enthält ein leeres Datenfeld.

2.3.4 Zusatzchunks

Neben den unverzichtbaren Chunks gibt es noch die *Zusatzchunks*. Sie enthalten weitere Informationen, die zur eigentlichen Darstellung des Bildes nicht unbedingt erforderlich sind. In Tabelle 2.3 sind die laut offizieller PNG-Spezifikation gültigen Zusatzchunks mit einer kurzen Funktionsbeschreibung aufgelistet. Man beachte, daß alle Chunks mit einem Kleinbuchstaben beginnen und an der zweiten Stelle einen Großbuchstaben stehen haben (siehe Abschnitt 2.3.2.2), es handelt sich also tatsächlich um *offizielle* (zweiter Buchstabe) *Zusatzchunks* (erster Buchstabe).

Detaillierte Informationen über die Chunks findet man in [Gro96a] und [Gro96b].

Typ	Bezeichnung	Funktion
bKGD	Background Color	Enthält eine Hintergrundfarbe
cHRM	Primary Chromaticities and White Point	Enthält Angaben für eine geräteunabhängige Farbdarstellung.
gAMA	Image Gamma	Enthält den Gamma-Wert des Bildes
hIST	Image Histogram	Enthält Häufigkeit der verwendeten Farben
pHYs	Physical Pixel Dimensions	Enthält phys. Größe des Bildes
sBIT	Significant Bits	Enthält Anzahl der ursprünglich verwendeten Bits
tEXt	Textual Data	Enthält textuelle Informationen
tIME	Image Last-Modification Time	Zeitangabe der letzten Änderung des Bildes
tRNS	Transparency	Enthält Angaben für einfache Transparenz
zTXt	Compressed Textual Data	Komprimierte Textdaten
oFFs	Image Offset	Bildposition auf einer Seite/Bildschirm/...
sCAL	Physical Scale of Image Subject	Maßstab des Bildes in Meter oder Grad
gIFg	GIF Graphic Control Extensions	Gewährt Kompatibilität zu GIF89a
gIFt	GIF Plain Text Extension	Gewährt Kompatibilität zu GIF89a
gIFx	GIF Application Extension	Gewährt Kompatibilität zu GIF89a
fRAc	Fractal Image Parameters	Enthält die Parameter, mit denen ein Fraktal-Bild erzeugt wurde.

Tabelle 2.3: Zusatzchunks

2.4 Kompression

2.4.1 Verwendeter Algorithmus

Der einzige Kompressionsalgorithmus, der bisher für das PNG-Format spezifiziert wurde, ist der Deflate/Inflate-Algorithmus von Phil Katz, der auch im Programm PKZIP verwendet wird ([DG96] und [Deu96]). Dabei werden in einem max. 32 KB großem Gleitfenster, das über die Eingabedaten geschoben wird, Datenmuster gesucht, die dann nach dem Huffman-Verfahren kodiert werden. Das Bild läßt sich danach ohne Verluste wieder rekonstruieren.

Diese Methode (ohne Huffman-Kodierung) wurde erstmals 1977 von Lempel und Ziv vorgestellt, wurde aber nicht patentiert. 1978 beschrieben sie dann eine einfachere Methode, um Muster in einem Text zu suchen. Terry Welch stellte daraufhin eine einfache Methode vor, diesen Algorithmus zu implementieren, was zu dem patentierten LZW-Kompressionsalgorithmus führte, der als Grundlage für die Kompression in GIF-Dateien diente.

2.4.2 Filter

Um die Kompression zu verstärken, kann ein Bild vor dem Kompressionsschritt gefiltert werden. Die Bilddaten sollen dadurch auf eine Weise repräsentiert werden, die für die Kompression besser geeignet ist.

Nach der bisherigen PNG-Spezifikation existiert nur eine Filtermethode, die fünf verschiedene Filtertypen umfaßt. Die Filterung wird jeweils auf die Bytes einer Bildzeile angewandt (unabhängig von Farbtyp, Bittiefe, ...). Der Filtertyp kann nach jeder Zeile gewechselt werden.

2.4.2.1 Typ 0: Kein Filter

Die Zeile wird unverändert an den Kompressionsalgorithmus übergeben.

2.4.2.2 Typ 1: Sub

Zu jedem Byte einer Zeile wird die Differenz dieses Bytes mit dem entsprechenden Vorgängerbyte berechnet.

$$Sub(x) = Raw(x) - Raw(x - bpp)$$

$Raw(x)$ bezeichnet hier die Bytes einer Zeile *vor* der Filterung. bpp ist die Anzahl Bytes pro Bildpunkt, bei Bildern mit weniger als 1 Byte pro Bildpunkt, wird dieser Wert auf 1 aufgerundet. Für ein RGB-Bild mit der Tiefe 8 Bit, erhält bpp beispielsweise einen Wert von 3. Es werden also immer die korrespondierenden Farbwerte subtrahiert.

2.4.2.3 Typ 2: Up

Der Up-Filter arbeitet prinzipiell genauso wie der Sub-Filter, mit dem Unterschied, daß nicht das linke Vorgängerbyte benutzt wird, sondern das Byte der Vorgängerzeile an der gleichen Position.

$$Up(x) = Raw(x) - Prior(x)$$

$Prior(x)$ bezeichnet die unmittelbar vorangegangene ungefilterte Bildzeile.

2.4.2.4 Typ 3: Average

Dieses Filtertyp berechnet die Differenz zum Durchschnitt von linkem und oberem Bildpunkt.

$$Average(x) = Raw(x) - \left\lfloor \frac{Raw(x - bpp) + Prior(x)}{2} \right\rfloor$$

$Raw(x)$, $Prior(x)$ und bpp wie bei Filtertyp 1 und 2.

2.4.2.5 Typ 4: Paeth

Der Paeth-Filter berechnet die Differenz zu *einem* der drei benachbarten Bildpunkte links, oben und links oben. Welcher Punkt tatsächlich verwendet wird, wird von einer Funktion abgeschätzt, die aus den drei Nachbarpunkten eine einfache lineare Funktion berechnet und denjenigen Punkt zurückgibt, der am nächsten an dem berechneten Wert liegt.

$$Paeth(x) = Raw(x) - Paeth_Schaezter(p_{links}, p_{oben}, p_{links_oben})$$

p_{links} ist dabei $Raw(x-bpp)$, p_{oben} ist $Prior(x)$ und p_{links_oben} $Prior(x-bpp)$. $Raw(x)$, $Prior(x)$ und bpp wie unter Typ 1 und 2 beschrieben. Diese Technik stammt von Alan W. Paeth, daher der Name.

In Abbildung 2.6 sind die verschiedenen Filtertypen anhand eines Beispiels dargestellt. Jeweils unter dem Bild ist das zugehörige Histogramm abgebildet, also die Häufigkeiten der einzelnen Graustufen.

2.5 Interlacing

2.5.1 Was ist Interlacing

Interlacing ist eine Methode, die es erlaubt, das Bild während des Betrachtens nach und nach aufzubauen. Dieses inkrementelle Anzeigen ist besonders dann sinnvoll, wenn das Laden des Bildes einige Zeit dauert, wie z.B. bei Netzwerk-Anwendungen wie dem World-Wide Web. Bei GIF ist es möglich, relativ schnell ein Bild in grober Auflösung darzustellen, das dann nach und nach während des Übertragens verfeinert wird. Handelt es sich bei diesem Bild um eine sogenannte „Imagemap“, bei der der Benutzer mit der Maus auf eine bestimmte Stelle klicken muß, reicht die erste grobe Auflösung oft schon aus, um zu erkennen, wo man die Maus hinbewegen muß.

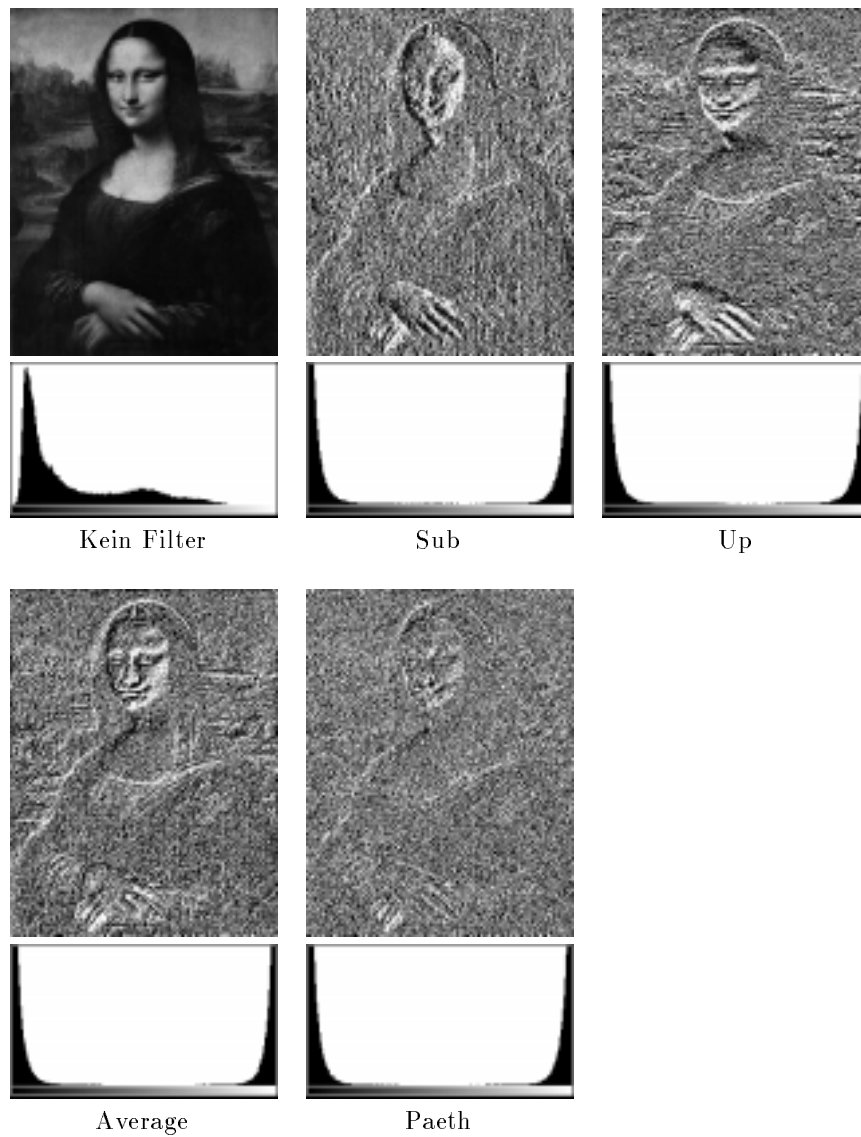


Abbildung 2.6: Vergleich der verschiedenen Filtertypen, mit Häufigkeiten der im Bild vorkommenden Graustufen.

2.5.2 Interlacing in GIF-Bildern

Das GIF-Format benutzt ein sehr einfaches Interlacing Schema, bei dem lediglich die Zeilen in einer anderen Reihenfolge übertragen werden als sie dargestellt werden. Das Schema beinhaltet vier Durchläufe: im ersten Durchlauf wird nur jede 8. Zeile übertragen (Zeile 1, 9, 17, ...), danach wird die Auflösung jeweils verdoppelt indem immer die Zeilen übertragen werden, die genau zwischen zwei schon übertragenen Zeilen liegen. Im zweiten Durchlauf sind das also die Zeilen 5, 13, 21, ... und im dritten die Zeilen 3, 7, 11, 15, Der vierte und letzte Schritt enthält dann alle restlichen Zeilen (2, 4, 6, 8, ...) und macht damit das Bild vollständig.



Abbildung 2.7: Die Stufen des GIF Interlace Schemas

Abbildung 2.7 zeigt die vier Durchläufe des GIF Interlacing Schemas. Hier sind auch gleich die beiden möglichen Arten des Darstellens abgebildet. In der jeweiligen linken Hälfte werden nur die bisher übertragenen Zeilen dargestellt, die restlichen bleiben ungesetzt (hier: Schwarz). Auf der rechten Seite werden die Zeilen dupliziert, so daß keine Lücken sichtbar sind.

2.5.3 Interlacing in PNG-Bildern

Das PNG-Format erlaubt es, mehrere Interlace-Schemata zu verwenden. Allerdings ist nach der bisherigen Spezifikation nur eine Methode definiert (es ist natürlich auch möglich überhaupt kein Interlacing zu verwenden). Die Methode arbeitet mit 7 Durchläufen und stammt von Adam M. Costello, daher ist die Methode auch als Adam7 bekannt.

Adam7 unterteilt das Bild in 8×8 Bildpunkte große Blöcke, innerhalb dieser Blöcke werden dann in jedem Durchlauf die in Abbildung 2.8 gekennzeichneten Punkte übertragen.

1	6	4	6	2	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7
3	6	4	6	3	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7

Abbildung 2.8: Adam7 Interlacing Schema

Die hinter Adam7 stehende Logik ist die, daß immer abwechselnd zunächst die horizontale Auflösung verdoppelt wird, dann die vertikale. Mit diesem Schema erhält man sehr schnell eine erste Version des Bildes, da im ersten Durchlauf nur $\frac{1}{64}$ des gesamten Bildes übertragen wird. Im Vergleich zu GIF lassen

sich kleine Schriften bei PNG etwa doppelt so schnell erkennen, nämlich nach etwa 25% übertragener Bilddaten, anstatt 50% bei GIF.

In Abbildung 2.9 sind die sieben Durchgänge einzeln dargestellt. Dabei sind die Beispiele wieder in zwei Hälften aufgeteilt. In der linken Hälfte werden nur die bisher übertragenen Bildpunkte dargestellt, während auf der rechten Seite die gesamten Teilblöcke mit der Farbe des Punktes gefüllt werden.

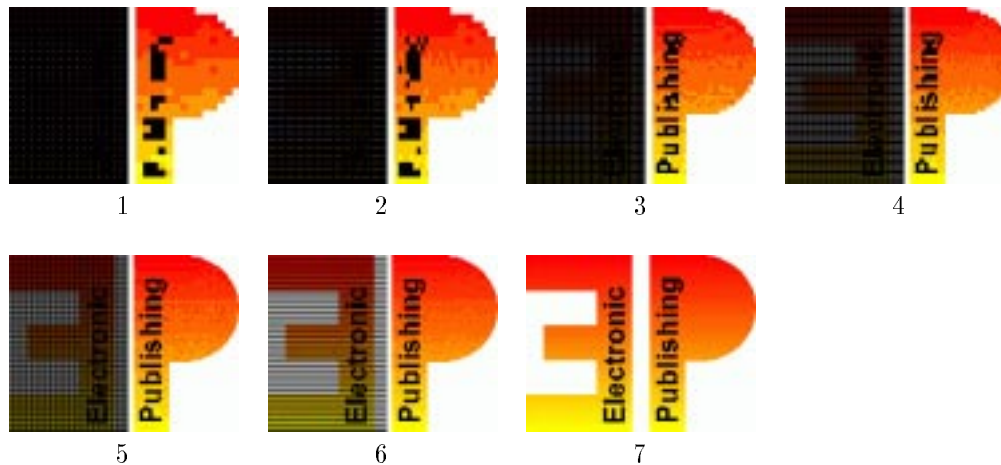


Abbildung 2.9: Die sieben Durchgänge des Adam7 Schemas

2.6 Transparenz

Das PNG-Format erlaubt es, Teile eines Bildes als transparent zu markieren. Wie diese Transparenz in den Dateien gehandhabt wird, soll in diesem Abschnitt kurz erläutert werden.

2.6.1 Was ist Alpha?

Wenn ein Bild an manchen Stellen durchsichtig erscheinen soll, müssen die Bildpunkte neben den eigentlichen Farbwerten noch ein weiteres Attribut erhalten, den *Alpha*-Wert. Dieser Wert regelt den Grad der Transparenz, wobei gilt: Bei einem Wert von 0 ist der Punkt völlig durchsichtig, während er beim Maximalwert vollkommen undurchsichtig ist. Dazwischenliegende Werte stehen dementsprechend für teilweise transparent, der Punkt muß dann mit dem Hintergrund kombiniert werden. Für den Alpha-Wert wird in der Regel die gleiche Anzahl Bits benutzt, wie für die einzelnen RGB-Werte bzw. wie für den Helligkeitswert bei Graustufenbildern. Es lassen sich also 2^{Bittiefe} unterschiedliche Transparenzstufen darstellen.

2.6.2 Einfache Transparenz

Bei der einfachen Transparenz wird einem bestimmten Farbwert genau *ein* Alpha-Wert zugeordnet. Haben zwei Bildpunkte die gleiche Farbe, so haben sie auch den gleichen Alpha-Wert. Diese Zuordnung *Farbe* \rightarrow *Alpha* wird in einem separaten Chunk, dem tRNS-Chunk, definiert.

Bei farbindizierten Bildern kann jedem Index der Farbtabelle ein eigener Alpha-Wert zugeordnet werden. Damit läßt sich bereits mehr als in GIF-Dateien realisieren. Bei GIF kann man lediglich *einen* Farbindex spezifizieren, dessen Bildpunkte dann als durchsichtig anzusehen sind (Alpha=0), alle anderen Punkte sind komplett undurchsichtig.

Auf Grauwert- bzw. Echtfarbenbildern läßt sich dieses Verfahren jedoch nicht anwenden, da dort keine Farbtabelle mehr vorhanden sind, sondern jeder Bildpunkt einen eigenen Farb- bzw. Helligkeitswert haben kann. In diesem Fall wird im tRNS-Chunk genau ein Grau- bzw. RGB-Wert angegeben, der dann automatisch einen Alpha-Wert von 0 erhält, während alle anderen Werte das Maximum erhalten (also vollkommen undurchsichtig sind). Diese Methode klingt ähnlich wie die bei GIF, der Unterschied ist jedoch, daß hier wirklich eine *Farbe* verloren geht, während bei GIF nur ein *Index* belegt werden muß, alle weiteren Indizes können jedoch immer noch mit allen zur Verfügung stehenden Farben besetzt werden. Das Verfahren reicht aber aus, um z.B. ein nicht-rechteckiges Bild zu erzeugen, indem der Randbereich mit der im tRNS-Chunk spezifizierten Farbe gefüllt wird.

2.6.3 Alpha-Kanal

Bei Grauwert- und Echtfarbenbildern ist die einfache Transparenz in vielen Fällen jedoch nicht mehr ausreichend, so kann man z.B. keine Alpha-Werte *zwischen* 0 und 1 vergeben und muß aufpassen, daß die „durchsichtige“ Farbe nicht im eigentlichen Bild selbst vorkommt, da dort nur Hintergrund zu sehen wäre. Abhilfe schafft in diesen Fällen der Alpha-Kanal.

Enthält ein Bild einen Alpha-Kanal, so wird zu jedem einzelnen Bildpunkt, zusätzlich zum eigentlichen Farbwert, ein Alpha-Wert gespeichert. Bei Echtfarbenbildern heißt das dann konkret, daß ein Bildpunkt nicht mehr nur durch ein RGB-Tripel beschrieben wird, sondern durch die vier Werte RGBA (Rot - Grün - Blau - Alpha). Entsprechend werden Punkte eines Graustufenbildes durch einen Grauwert und einen Alpha-Wert gekennzeichnet.

Der Alpha-Kanal kann somit als Transparenz-Maske angesehen werden. Im Unterschied zur einfachen Transparenz ist hier das ursprüngliche Bild noch komplett enthalten, d.h. selbst an den Stellen an denen das Bild nicht sichtbar ist, ist noch die volle Bildinformation in der Datei gespeichert. Man kann also nachträglich den Alpha-Kanal wieder entfernen und so das gesamte Ausgangsbild wieder sichtbar machen.

In Abbildung 2.10 wird die Anwendung eines Alpha-Kanals illustriert. Bild (b) erhält dabei den in (c) abgebildeten Alpha-Kanal und wird gegen den Hintergrund (a) dargestellt, das Ergebnis ist (d). Man beachte, daß (b) und (c) in *einer* PNG-Datei gespeichert werden. Der Hintergrund muß allerdings von der Anwendung kommen, die das Bild auf dem Bildschirm darstellt.

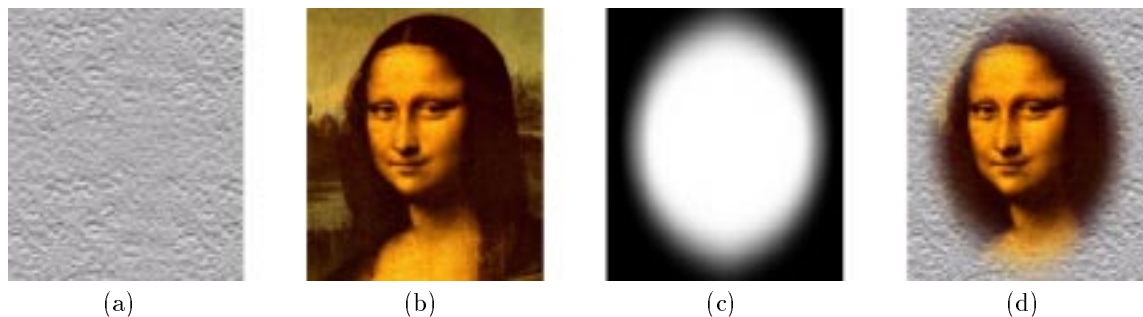


Abbildung 2.10: Beispiel für die Anwendung eines Alpha-Kanals

2.7 Farbwiedergabe

In diesem Kapitel soll auf eine PNG-Eigenschaft eingegangen werden, die es ermöglicht, die Farben eines Bildes auf verschiedenen Systemen in gleicher Weise darzustellen. Beispielsweise wird ohne Korrektur ein und dasselbe Bild auf einem PC dunkler dargestellt als auf einem Macintosh. Ein Grafikformat müßte

also Informationen enthalten, mit denen das Bild auf jedem Rechnersystem in gleicher Weise dargestellt werden kann. Dies läßt sich bereits mit einem einzigen Wert recht gut erreichen, dem *Gamma*-Wert.

Die verschiedenen Anzeigegeräte stimmen darüber überein, daß der Wert 0 für Schwarz und 255 für Weiß steht, jedoch herrscht keine Übereinkunft für welche Intensität der Wert 128 steht. Bei Geräten mit linearem Intensitätsverläufen steht es wirklich für 50%, bei anderen Geräten kann es jedoch auch über 70% bedeuten. Die tatsächlichen Verläufe lassen sich recht gut durch eine einfache Potenzfunktion approximieren:

$$\text{Ausgabe} = \text{Eingabe}^\gamma$$

Die Ein- und Ausgabe muß hierbei auf den Bereich zwischen 0 und 1 normiert sein. Der Exponent ist nun der Gamma-Wert des entsprechenden Gerätes (eine ausführlichere Beschreibung zu Gamma findet man in [Poy95]).

Kennt man also den Gamma-Wert des eigenen Rechnersystems und den Gamma-Wert des Systems, auf dem das Bild erzeugt wurde, können die Farbwerte so angepaßt werden, daß das Bild auf beiden Systemen etwa gleich dargestellt wird. Der zweitgenannte Gamma-Wert ist derjenige, der in einer PNG-Datei gespeichert werden kann.

Allerdings muß sich das Programm, das ein Bild darstellen möchte, selbst um die Farbkorrektur kümmern. Es muß also auf irgendeine Weise den eigenen systemabhängigen Gamma-Wert erhalten haben, um das Bild korrekt darstellen zu können, z.B. durch eine Eingabe des Benutzers.

Da mit dem Gamma-Wert das Ursprungsbild nur näherungsweise rekonstruiert werden kann, gibt es für höhere Ansprüche noch die Möglichkeit, die Farben nach dem CIE-Modell (Commission Internationale de l'Éclairage, 1931) zu spezifizieren.

2.8 Zusammenfassung

Alles in allem kann man sagen, daß das PNG-Format, was das Speichern einzelner Bilder angeht, dem GIF-Format in allen Punkten überlegen ist. Die wesentlichen Verbesserungen seien hier noch einmal kurz erwähnt:

Echtfarben- und Graustufenbilder: Durch diese Option lassen sich Bilder nun wirklich „verlustlos“ speichern.

Bessere Kompression: PNG-Dateien sind in der Regel kürzer als Dateien anderer vergleichbarer Formate (siehe auch Tabelle 2.4).

Transparenz vielseitiger: Mit dem Alpha-Kanal hat man volle Kontrolle über die Transparenz jedes einzelnen Bildpunktes.

Inkrementelles Anzeigen schneller: Für das erste Rohbild müssen nur $\frac{1}{64}$ der Bilddaten übertragen werden (bei GIF ist es $\frac{1}{8}$).

Farbkorrektur wird ermöglicht: Durch den Gamma-Wert und die Möglichkeit, Farben nach dem CIE-Modell zu speichern, ist die Voraussetzung geschaffen, Bilder auf verschiedenen Systemen in gleicher Weise darstellen zu können.

Beschädigte Dateien werden erkannt: Fehlerhafte Dateien können anhand der Signatur und anhand den CRC-Werten in den Chunks schnell und zuverlässig erkannt werden.

Leicht erweiterbar: Es können leicht eigene Chunks zugefügt werden, ohne daß dadurch die Portabilität verloren geht. Die neuen Chunks werden von den bisher existierenden Programmen einfach überlesen.

Quelltexte verfügbar: Es existieren bereits frei verfügbare C-Bibliotheken zum Lesen und Schreiben von PNG-Dateien (libpng/zlib).

	PNG	GIF	JPEG (verlustlos)	TIFF (LZW/Pred)
256 Farben	1,00	1,21	-	1,37
Graustufen	1,00	1,46	1,02	1,47
Echtfarben	1,00	-	1,29	1,38

Tabelle 2.4: Dateigrößen im Verhältnis zu PNG-Dateien

GIF bietet allerdings eine Option, die im PNG-Format nicht enthalten ist, nämlich das Speichern mehrerer Bilder in einer Datei. Damit lassen sich dann kleine Animationen realisieren, die inzwischen auf vielen WWW-Seiten zu sehen sind. Das PNG-Format ist jedoch strikt als ein Einzelbild-Format konzipiert und wird diese Option auch in Zukunft nicht anbieten. Grund dafür ist, daß das Format einfach und austauschbar sein soll. Würde man mehrere Bilder in einer Datei erlauben, gäbe es dennoch viele Programme, die diese Option nicht unterstützen würden, so wie es ja heute beim GIF-Format der Fall ist. Die Datei wäre dann also doch nicht mehr einfach austauschbar.

Eine Erweiterung auf mehrere Bilder ist dennoch nicht ganz ausgeschlossen, allerdings wird diese Erweiterung einen neuen Namen und damit auch eine neue Signatur erhalten, es handelt sich dann also nicht mehr um eine reine *PNG*-Datei. Ein Programm kann dann selbst entscheiden, ob es nur Einzelbilder oder auch Bildsequenzen unterstützen möchte.

Ein weiterer Nachteil von PNG ist, daß es sich bisher (noch) nicht gänzlich als Standard durchgesetzt hat. Es gibt zwar inzwischen viele Programme, die PNG unterstützen, vor allem im Shareware-Bereich (z.B. Paint Shop Pro, Graphics Workshop, POVray, ...), so daß man bereits Bilder konvertieren und bearbeiten kann, jedoch werden nur selten alle Möglichkeiten von PNG ausgeschöpft (Gamma-Korrektur, Alpha-Kanal).

Allerdings hat es den Anschein, daß die bestehenden Programme (auch kommerzielle Software) jetzt gerade nach und nach auf PNG-Unterstützung aufgerüstet werden. Die meisten Web-Browser sollen beispielsweise in den kommenden Versionen PNG schon im Programm eingebaut haben (Mosaic, Internet Explorer), während andere das Format nur durch PlugIns unterstützen möchten (Netscape).

Alles in allem kann man jedoch sagen, daß PNG zur Speicherung von Einzelbildern eine bessere Alternative als GIF ist, da alle GIF-Optionen auch in PNG enthalten sind und in der Regel sogar übertroffen werden. Es bleibt also abzuwarten, ob es wirklich zu dem Standard werden wird, der GIF bisher war.

PNG-Homepage

Die neuesten Informationen zu PNG, sowie die komplette PNG-Spezifikation, Beispiel-Bilder, C-Quelltexte, ... usw. befinden sich im Internet auf der PNG-Homepage, die unter folgenden Adressen zu erreichen ist:

<http://www.wco.com/~png>

oder

<http://shire.ncsa.uiuc.edu/PNG>

2.9 Literatur

- [Deu96] P. Deutsch. *DEFLATE Compressed Data Format Specification version 1.3*. Aladdin Enterprises, Mai 1996.

- [DG96] P. Deutsch und J-L. Gailly. *ZLIB Compressed Data Format Specification version 3.3*. Aladdin Enterprises, Mai 1996.
- [Gro96a] PNG Group. *PNG (Portable Network Graphics) Specification Version 1.0*. URL: <http://www.wco.com/~png/pngdocs.html>, 10.06.1996.
- [Gro96b] PNG Group. *PNG Special-Purpose Public Chunks*. URL: ftp://ftp.uu.net/graphics/png/documents/pngextensions.*, 11.03.1996. Dokument in mehreren Formaten verfügbar.
- [Int94] International Telecommunications Union. *Error-correcting Procedures for DCEs Using Asynchronous-to-Synchronous Conversion*, 1994. ITU-T Recommendation V.42, Rev. 1.
- [IS 84] International Organization for Standardization. *Information Processing Systems - Data Communication High-Level Data Link Control Procedure - Frame Structure*, 3. Auflage, Oktober 1984.
- [Poy95] C. Poynton. *Gamma FAQ*. URL: <http://www.inforamp.net/~poynton/Poynton-colour.html>, 28.05.1995.

Kapitel 3

Virtual Reality Modeling Language (VRML): Interaktive 3D-Dokumente

Jürgen Moßgraber

3.1 Übersicht

Die Virtual Reality Modeling Language (VRML) ist ein Dateiformat für die Beschreibung von interaktiven dreidimensionalen Welten und Objekten. Sie kann benutzt werden in Verbindung mit dem World Wide Web. Mit VRML können dreidimensionale Repräsentationen von komplexen Szenen wie Illustrationen, Produktdefinitionen und Präsentationen virtueller Realitäten erzeugt werden.

3.2 Einleitung

Die *Virtual Reality Modeling Language (VRML)* ist eine Sprache für die Beschreibung von interaktiven Simulationen mit mehreren Teilnehmern: Über das weltweite Internet verteilte virtuelle Welten sind verbunden über das *World Wide Web*¹. VRML ist fähig, statische und bewegte Objekte darzustellen. Es kann *Hyperlinks* zu anderen Medien, wie z.B. Klänge, Filme und Bilder enthalten. Im Gegensatz zu Programmiersprachen wie z.B. C++, muß VRML nicht in Maschinensprache übersetzt werden. VRML Dateien werden interpretiert und dann dargestellt. Interpreter (Browser) sind für viele Rechnerplattformen erhältlich und auch an Autorenumgebungen für die Erstellung von VRML-Dateien herrscht kein Mangel. VRML unterstützt ein Erweiterungsmodell, das es erlaubt, neue Elemente zu definieren. Außerdem gibt es eine Schnittstelle für VRML Elemente und die gängigsten 3D Anwendungsschnittstellen (API).

3.3 Entstehungsgeschichte

3.3.1 VRML 1.0

Mark Pesce und Tony Parisi entwickelten 1993 eine dreidimensionale Schnittstelle zum World Wide Web, die sie im Frühjahr 1994 auf der ersten WWW-Konferenz in Genf vorstellten. Dort kam auch der Anstoß zur Entwicklung der VRML-Sprache von einem Mitbegründer des Web Tim Berners-Lee und Dave Ragget. Sie waren die Initiatoren einer Arbeitsgruppe, die sich seitdem um den Designprozeß von VRML bemüht. Dieser entwickelte sich in Folge in einer für das Web typischen Arbeitsweise: Mit Hilfe des amerikanischen Magazins „Wired“ wurde eine *Electronic Mailing List* initiiert, die nach kürzester Zeit über 1000 Mitglieder zählte, und in der sich nach intensiver Diskussion ein Konsens über die Anforderungen an eine VRML fand.

Da man das Rad nicht neu erfinden wollte, sah man sich nach existierenden Dateiformaten für dreidimensionale Szenen um und einigte sich dann auf das *Open Inventor File Format* der Firma Silicon

¹Das World Wide Web (WWW) ist ein Projekt, das die globale, gemeinsame Nutzung von Informationen zum Ziel hat. Ursprünglich am CERN (dem Kernforschungszentrum in Genf) entwickelt, verbreitet es sich mit rasanter Geschwindigkeit.

Graphics. Eine Teilmenge dieses Formats, mit Netzwerk Erweiterungen versehen, bildete dann die Basis von VRML.

3.3.2 VRML 2.0 (Moving Worlds)

Um die technische Entwicklung voranzutreiben, ergriffen acht VRML-Pioniere, darunter Mark Pesce, Gavin Bell und Tony Parisi die Initiative und gründeten zwei Wochen nach einem Treffen auf der SIGGRAPH 95 (Herbst 1995) die „VRML Architecture Group“ (VAG), die sich seitdem als eine Art oberstes technisches Lenkungsgremium der Entwicklung von VRML widmet.

Mit Hilfe der *Mailing List* arbeitete die VAG dann an der Version 1.1, die Erweiterungen wie Interaktivität enthalten sollte. Dazu sollten z.B. Funktionen wie Kollisionsvermeidung in VRML integriert werden, um das Interagieren von Avataren² zu erleichtern.

Die Version 1.1 wurde dann aber zugunsten von 2.0 fallen gelassen. Dazu kam es, nachdem nun auch große Unternehmen an der Entwicklung teilnehmen wollten. Für die Version 2.0 wurden dann mehrere Vorschläge bei der VAG eingereicht, über die in virtuellen Wahlkabinen abgestimmt wurde.

Folgende Vorschläge gab es:

1. *Moving Worlds* von SGI und Sony, in den die Erweiterungen von 1.1 vollständig einfließen
2. Sun versuchte mit *HoloWeb* größeren Einfluß auf die Integration von VRML und Java zu gewinnen
3. *ActiveVRML* war ein von Microsoft recht spät eingereichter Vorschlag der auf DirectX aufbauen sollte
4. *Reactive Virtual Enviroment* von IBM Japan
5. *Out Of This World* von Apple basierend auf dem QuickDraw 3D-Metafile-Format
6. *Dynamic Worlds* der deutschen GMD, St.Augustin wollte auf den Aspekt der Multi-User-Nutzung aufmerksam machen

Die Wahl fiel auf *Moving Worlds*. Die internationale Standardbezeichnung lautet ISO³/IEC⁴ WD 14772.

3.4 Wozu braucht man VRML ?

Das World Wide Web bietet eine Abstraktion des Internets. Diese Abstraktion besitzt ein *Adressierungsschema*, einen eindeutigen Identifizierer (*Univereal Resource Locator*: URL), der jedem sagt „wo er hingehen kann und wie er dort hinkommt“. Das ist die einzige Navigation, die möglich ist (außer dem *Surfen*⁵). Obwohl dies für Nicht-Experten vieles vereinfacht, fehlt es an Dimensionalität.

Die Mehrheit der Menschen wird das Web nur benutzen, wenn es sich wie die richtige Welt verhält. Abstrakte URLs sind schwer zu merken und das Springen von einem Textfragment zum nächsten ist keine Tätigkeit, die von Menschen täglich durchgeführt wird.

Was also benötigt wird, ist ein *Design der Desorientierung*, und dies soll mit Hilfe von VRML verwirklicht werden.

3.5 Was kann VRML ?

Es wurde Wert darauf gelegt, daß folgende Grundvoraussetzungen erfüllt sind:

1. Möglichkeit der Entwicklung von Anwendungsgeneratoren und -editoren und die Importierbarkeit von anderen Industrieformaten

²Visuelle Repräsentationen von Personen in einer virtuellen Welt.

³International Organization for Standardization

⁴International Electrotechnical Comission

⁵Anwählen von Hyperlinks die sich auf der aktuellen WWW-Seite befinden, um so eine andere Seite aufzurufen.

2. Bekanntgabe aller Informationen, die für die Implementation nötig sind und Angebot einer vollständigen Funktionsmenge für breite Akzeptanz in der Industrie
3. Kombinierbarkeit von VRML-Elementen, um so die Wiederverwendbarkeit zu unterstützen
4. Beliebige Erweiterbarkeit der Sprache
5. Unabhängigkeit von der verwendeten Hardware-Plattform
6. Mehrbenutzerbetrieb
7. Die VRML Elemente sollten unabhängig von einander sein oder alle Abhängigkeiten müssen strukturiert und wohldefiniert sein (Orthogonalität)
8. Die Benutzbarkeit mit langsamen Netzverbindungen (14400-Bit/s-Modem) soll noch hinreichend möglich sein.

Der letzte Punkt wird dadurch erfüllt, daß VRML-Dateien sehr klein sind, da sie im Gegensatz zu Bild- oder Videodateien eine Szene nur beschreiben, anstatt jeden Bildpunkt zu speichern.

Da durch diese Methode die VRML-Datei auf dem Zielrechner interpretiert und somit die Welt berechnet werden muß, ist nun nicht mehr die Übertragungsleistung der Flaschenhals, sondern die Rechengeschwindigkeit des Zielrechners.

Um diesem Problem wiederum zu begegnen, gibt es verschiedene Detaillierungsstufen der Darstellung, die bei Bedarf nachgeladen werden (*inlining*). Dabei wird ausgenutzt, daß entferntere Objekte detailärmer dargestellt werden können. Unterschreitet der Betrachter einen gewissen Abstand zum Objekt wird die nächste Detaillierungsstufe nachgeladen.

3.5.1 Die Möglichkeiten von VRML 1.0

Die erste Version von VRML erlaubt die Erzeugung von virtuellen Welten mit beschränktem interaktivem Verhalten. In ihnen können Objekte enthalten sein, die auch als *Hyperlinks* fungieren können. Diese können wiederum auf eine VRML-Welt verweisen oder auf HTML-Dokumente und andere gültige *MIME-Typen*. Wenn der Benutzer ein Objekt mit einem Hyperlink auswählt, wird das Programm gestartet, das dem gewählten MIME-Typ zugeordnet ist und ihn darstellt.

3.5.2 Die Möglichkeiten von VRML 2.0

VRML 2.0 bietet folgende Ergänzungen und Erweiterungen :

Erweiterte statische Welten Die virtuellen Welten können jetzt realistischer gestaltet werden. Neue Nodes erlauben es, Hintergründe wie Boden, Himmel, entfernte Berge, Wolken und Nebel zu Szenen hinzuzufügen. Außerdem besteht die Möglichkeit, Klänge in Szenen einzufügen.

Interaktion Man kann nun direkt mit Objekten oder Wesen, die man trifft, interagieren und sie bewegen. *Sensornodes* lösen Ereignisse aus, wenn man bestimmte Abschnitte der Welt betritt oder Objekte anwählt. Auch Kollisionsprüfungen sind nun möglich, um die Durchdringung von Mauern zu verhindern.

Animation Es gibt eine Vielzahl von Animationsobjekten, die *Interpolator* genannt werden. Sie ermöglichen fliegende Vögel, automatisch öffnende Türen oder laufende Roboter. Auch können Objekte ihre Farbe ändern, wenn sie sich bewegen, wie z.B. die Sonne.

Einbindung von Skripten und Programmiersprachen Skripten erlauben es nicht nur Wesen und Objekte zu bewegen, sondern ihnen auch den Anschein von Intelligenz zu geben.

Prototypen Mehrere Nodes können zu einem Prototypen zusammengefaßt werden, der dann als neue Node genutzt werden kann.

3.6 Spezifikation

3.6.1 Grundlagen der Sprache

Beginnen wir zunächst mit einem Beispiel. Der folgende Text erzeugt eine Szene mit einer roten Kugel und einem blauen Würfel:

```
#VRML V2.0 utf8
Transform
{
    # Wir machen unser eigenes Licht
    children [ NavigationInfo { headlight FALSE }
              # Licht das die Welt erleuchtet
              DirectionalLight { direction 0 0 -1 }

    Transform
    {
        translation 3 0 1
        children [ Shape
            {
                geometry Sphere { radius 2.3 }

                appearance Appearance
                {
                    material Material
                    {
                        # Rot
                        diffuseColor 1 0 0
                    }
                }
            }
        ]
    }
}
Transform
{
    translation -2.4 .2 1
    rotation 0 1 1 .9
    children [ Shape
        {
            geometry Box {}

            appearance Appearance
            {
                material Material
                {
                    # Blau
                    diffuseColor 0 0 1
                }
            }
        }
    ]
}
}
```

Was man aus dem Beispiel erkennen kann ist, daß jede VRML-Datei mit der Zeile

```
#VRML V2.0 utf8
```

beginnen muß, wobei *utf8* den verwendbaren Zeichensatz angibt. UTF-8 (Unicode) ist beschrieben in ISO 10646.

Das „#“ Zeichen leitet einen Kommentar ein, alle Zeichen bis zur nächsten Zeile werden ignoriert⁶. Die einzige Ausnahme ist in *Strings*⁷, in denen es als Teil des Strings betrachtet wird.

VRML unterscheidet Groß-/Kleinschreibung, *sphere* ist also ungleich *Sphere*.

Nach der Kopfzeile kann jede Kombination folgender Typen vorkommen:

- Eine beliebige Anzahl von *Prototypen*
- Eine beliebige Anzahl von *Children Nodes*
- Eine beliebige Anzahl von *Routes*

Was Prototypen, Children Nodes und Routes sind, wird weiter unten erklärt.

Die Dateierweiterung für VRML Dateien ist *.wrl* (für *world*). Der MIME Typ für VRML 1.0 ist **x-world/x-vrml** und für 2.0 **model/vrml**.

3.6.2 Das Koordinatensystem

VRML benutzt ein kartesisches, rechtsorientiertes, dreidimensionales Koordinatensystem.⁸ Standardmäßig werden Objekte auf den zweidimensionalen Bildschirm abgebildet, indem sie in Richtung der positiven Z-Achse, mit der positiven X-Achse zur Rechten und der positiven Y-Achse nach oben projiziert werden.

3.6.3 Die Struktur des Szenengraphen

Auf der höchsten Abstraktionsebene ist VRML nur eine Möglichkeit, um Objekte lesen und schreiben zu können. Theoretisch können diese Objekte alle Arten von Daten enthalten (3D Geometriedaten, MIDI-Daten, JPEG-Bilder, usw.).

VRML definiert eine Menge von Objekten, die nützlich für 3D-Grafiken sind. Diese Objekte heißen *Nodes (Knoten)*.

Diese sind in hierarchischen Strukturen angeordnet, die *Szenengraphen* (scene graphs) genannt werden. Nodes die früher in einem Szenengraphen vorkommen, können sich auf spätere auswirken. Bsp.: *Rotations* oder *Material Nodes*.

Eine Node besitzt die folgenden Eigenschaften und Informationen:

- *Um welche Art von Objekt es sich handelt*: Eine Kugel, ein Würfel, eine Transformation, Texturdaten, usw.
- *Die Parameter (Fields) der Node*: Zum Beispiel der Radius einer Kugel
- *Eine Menge von zugeordneten Ereignissen (Events)*: Nodes können Events empfangen (**eventIn** oder **set_events**), die typischerweise die Node ändern (*set_position*, *set_color*, usw.). Events können auch gesendet werden (**eventOut** oder **of_changed**), die anzeigen, daß sich etwas in der Node geändert hat (*position_changed*, *color_changed*, usw.). Das Schlüsselwort *exposedField* kann als Abkürzung benutzt werden, wenn ein Feld mit einem Aus- und Eingabeevent verbunden ist:

```
exposedField banane
```

ist äquivalent zu

```
eventIn set_banane
```

```
field banane
```

```
eventOut banane_changed
```

⁶Kommentare und Leerzeichen werden nicht übertragen um Bandbreite einzusparen. Um zugehörige Informationen übertragen zu können, sollte man deshalb *Info Nodes* benutzen.

⁷Zeichenketten

⁸Drei Zahlengeraden, die paarweise aufeinander senkrecht stehen und auf denen mit der gleichen Einheit gemessen wird.

Die Syntax für eine Node ist folgende:

Nodetyp { **Felder** }

Nur der Nodetyp und die geschweiften Klammern sind notwendig der Rest ist optional. Felder, die nicht angegeben werden, sind mit Standardwerten belegt.

3.6.3.1 Instantiierung

Auch eine Node kann in einer VRML-Datei mehrmals benutzt werden. Dies heißt Instantiierung. Das Schlüsselwort DEF definiert den Namen einer Node und erzeugt eine Node dieses Typs. Mit USE kann man dann diese Node wieder benutzen. Wird eine dieser Nodes geändert, ändern sich alle Nodes. Ein Name kommt in den Gültigkeitsbereich, sobald der Interpreter auf das Schlüsselwort DEF trifft und bleibt darin solange, bis ein anderes DEF mit dem gleichen Namen auftritt oder das Dateiende erreicht ist. USE kann nicht auf einen Namen angewandt werden, der sich in einer anderen Datei befindet.

Beispiel:

```
#VRML V2.0 utf8
Transform
{
  children [ DEF Joe Shape
    {
      geometry Sphere {}
    }

    Transform
    {
      translation 2 0 0
      children DEF Joe Shape
        {
          geometry Sphere
            {
              radius .2
            }
        }
    }

    Transform
    {
      translation -2 0 0
      children USE Joe
    }
  ]
}
```

3.6.4 Felder (*fields*)

Es gibt zwei Klassen von Feldern:

- Felder, die einen einzelnen Wert enthalten (wobei ein Wert z.B. eine Zahl, ein Vektor oder sogar ein Bild sein kann)
- Felder, die mehrere Werte enthalten

Felder, die nur einen Wert enthalten, beginnen alle mit „SF“. Mehrwertige Felder haben Namen die mit „MF“ anfangen. Mehrwertige Felder werden als Folge von Werten, die durch *Whitespaces* (z.B. Kommas) getrennt sind, geschrieben und in eckige Klammern eingeschlossen.

Es gibt die folgenden einwertigen und mehrwertigen Felder:

- **SFBool:** Boolesche(-r) Wert(-e). Bsp.:

```
einZustand TRUE
```

- **SFColor/MFColor:** Farbwert(-e) bestehen aus je drei Gleitkommazahlen (RGB) im Bereich von 0.0 bis 1.0. Bsp. (Die Farben Rot, Grün und Blau):

```
someColors [ 1.0 0. 0.0, 0 1 0, 0 0 1]
```

- **SFFloat/MFFloat:** Gleitkommawert(-e) im ANSI C Format. Bsp.:

```
someFloats [ 3.1415926, 12.5e-3, .0001 ]
```

- **SFImage:** Unkomprimiertes 2-dimensionales Farb- oder Schwarzweißbild. Angegeben werden die Breite, die Höhe, die Anzahl der Komponenten ⁹ und die einzelnen Pixelwerte von links-unten nach rechts oben. Bsp.:

```
einBild 2 4 3 0xFF0000 0xFF00 0 0 0 0 0xFFFFF 0xFFFF00
          # Rot      Gruen  Schwarz Weiss    Gelb
```

Das Ergebnis sieht folgendermaßen aus:

Weiß	Gelb
Schwarz	Schwarz
Schwarz	Schwarz
Rot	Grün

- **SFInt32/MFInt32:** 32-Bit Wert(-e). Bsp.:

```
einigeWerte [ 17, -0xE20, -518820 ]
```

- **SFNode/MFNode:** VRML Node(-s). Beispiel mit 4 Nodes:

```
vierNodes [ Transform { translation 1 0 0 }
            DEF CUBE Box {}
            USE CUBE
            USE SOME_OTHER_NODE ]
```

- **SFRotation/MFRotation:** Beliebige rechtsorientierte Rotation(-en) um einen Rotationsvektor. Angegeben werden 3 SFFloats für den Rotationsvektor und dann der Winkel in Radianen. Bsp. (Drehung um 180 Grad um die Y-Achse):

```
eineRotation 0.0 1.0 0.0 3.14159265
```

- **SFString/MFString:** UTF-8 Zeichenkette(-n). Um Hochkommatas auszugeben, muß ein *Backslash* vorangestellt werden. Für einen Backslash benötigt man dann zwei Backslashes. Bsp.:

⁹Ein Bild mit einer Komponente besteht aus hexadezimalen Werten (je 1 Byte), die die Intensität des Bildes darstellen. 0xFF ist also die höchste Intensität und 0x00 bedeutet keine Intensität. Ein Bild mit zwei Komponenten hat die Intensität im ersten Byte (high) und die Transparenz im zweiten (low) Byte. Pixels in einem Dreikomponenten Bild haben die Rotkomponente im ersten Byte, gefolgt von Grün und Blau (0xFF0000 ist also Rot). Vierkomponenten Bilder haben das alpha Byte nach Rot/Grün/Blau (0x0000FF80 ist halbtransparentes Blau). Ein Wert von 0x00 ist total transparent, 0xFF ist undurchsichtig.

```
einText [ " \" Ersma guten ", "Tach hier ! \" " ]
```

- **SFTime/MFTime:** Zeitangabe(-n). Anzahl der Sekunden (Gleitkommazahl mit doppelter Genauigkeit) nach dem 1.1.1970 00:00:00 GMT.

- **SFVec2f/MFVec2f:** Zweidimensionale(-r) Vektor(-en) angegeben als Gleitkommazahlen. Bsp.:

```
einVektor [ 42 666, 7.7 10 ]
```

- **SFVec3f/MFVec3f:** Dreidimensionale(-r) Vektor(-en) angegeben als Gleitkommazahlen. Bsp.:

```
nochEinVektor [ 1 42 666, 7.7 9 10 ]
```

Die Standardeinheiten für Längen und Abstände sind Meter.

3.6.5 Events

Die meisten Nodes haben mindestens einen *eventIn* definiert und können somit *events* empfangen. Empfangene events sind Nachrichten, die Daten enthalten, die von anderen Nodes abgeschickt wurden. Sie können den Zustand der Node, die die Nachricht empfängt, ändern.

Einige Nodes besitzen *eventOut* Definitionen. Diese werden benutzt, um Datenmeldungen an Zielnodes zu schicken um anzuzeigen, daß sich im Absender etwas geändert hat.

3.6.5.1 Routes

Die Verbindung zwischen einer Node, die einen Event generiert und einer Node, die ihn empfängt, wird als *route* bezeichnet. Damit die Verbindung möglich ist, muß die Typen des *eventIn* und des *eventOut* passen. Es ist also nicht erlaubt, einen *SFFloat* auf einen *SFInt32* zu *routen*. Die Syntax für eine Route ist folgende:

```
ROUTE NodeName.eventOutName_changed TO NodeName.set_eventInName
```

Hierzu ein Beispiel:

```
DEF CLICKER TouchSensor { enabled TRUE }
DEF LIGHT DirectionalLight { on FALSE }
ROUTE CLICKER.enabled TO LIGHT.on
```

oder

```
ROUTE CLICKER.enabled_changed TO LIGHT.set_on
```

Schickt eine Node einen *eventOut* ab Aufgrund eines empfangenen *eventIn*, so können *Eventkaskaden* entstehen. Diese Events enthalten alle den gleichen Zeitstempel, um möglicherweise entstehende Schleifen abbrechen zu können.

3.6.5.2 Sensoren

Sensoren Nodes erzeugen Events. Man unterscheidet zwei Varianten:

- **Geometrische Sensornodes:** Erzeugen Events, die auf einer Aktion des Benutzers beruhen (z.B. Mausklick)
- **Zeitliche Sensornodes:** Senden Events in bestimmten Zeitintervallen

Der *Browser* kontrolliert den Zeitablauf durch die Abgabe von Events der Zeitsensoren. Es ist nicht festgelegt daß dieser Ablauf der Echtzeit entsprechen muß.

3.6.6 Nodes (Knoten)

Zunächst soll eine Übersicht über alle Nodes gegeben werden die in 9 Klassen eingeteilt sind (Tabelle 3.1 und 3.2).

Im folgenden wird jede Node kurz erklärt und ihre Syntax beschrieben. Ausführliche Beschreibungen findet man z.B. in [GB96].

3.6.6.1 Anchor

```
Anchor
{
    eventIn      MFNode    addChildren
    eventIn      MFNode    removeChildren
    exposedField MFNode    children      []
    exposedField SFString  description   ""
    exposedField MFString  parameter     []
    exposedField MFString  url           []
    field        SFVec3f   bboxCenter    0 0 0
    field        SFVec3f   bboxSize      -1 -1 -1
}
```

Die Gruppierungsnode Anchor bewirkt, daß eine URL übers Netz geladen wird wenn das Anzeigeprogramm die Geometrie, die durch die Kinder der Anchor Node beschrieben wird, darstellt.

3.6.6.2 Appearance

```
Appearance
{
    exposedField SFNode material      NULL
    exposedField SFNode texture       NULL
    exposedField SFNode textureTransform NULL
}
```

Die Appearance Node legt die visuellen Eigenschaften einer Geometrie fest durch die Angabe von Material- und Texturnodes.

3.6.6.3 AudioClip

```
AudioClip
{
    exposedField SFString  description   ""
    exposedField SFBool    loop          FALSE
    exposedField SFFloat   pitch         1.0
    exposedField SFTIME    startTime     0
    exposedField SFTIME    stopTime      0
    exposedField MFString  url           []
    eventOut      SFTIME    duration_changed
    eventOut      SFBool    isActive
}
```

AudioClip enthält Klangdaten, die von anderen Nodes verwendet werden können.

3.6.6.4 Background

```
Background
{
```

Gruppierungen (Grouping Nodes)	Anchor Billboard Collision Group Transform
Sensoren (Sensors)	CylinderSensor PlaneSensor ProximitySensor SphereSensor TimeSensor TouchSensor VisibilitySensor
Aussehen (Appearance)	Appearance FontStyle ImageTexture Material MovieTexture PixelTexture TextureTransform
Spezielle Gruppierungen (Special Groups)	Inline LOD Switch
Geometrie (Geometry)	Box Cone Cylinder ElevationGrid Extrusion IndexedFaceSet IndexedLineSet PointSet Sphere Text
Interpolatoren (Interpolators)	ColorInterpolator CoordinateInterpolator NormalInterpolator OrientationInterpolator PositionInterpolator ScalarInterpolator
Allgemeine Nodes (Common Nodes)	AudioClip DirectionalLight PointLight Script Shape Sound SpotLight WorldInfo

Tabelle 3.1: Übersicht über alle Nodes (a)

Geometrische Eigenschaften (Geometric Properties)	Color Coordinate Normal TextureCoordinate
Bindable Nodes (Nodes von denen jeweils nur eine aktiv sein kann)	Background Fog NavigationInfo Viewpoint

Tabelle 3.2: Übersicht über alle Nodes (b)

```

eventIn      SFBool    set_bind
exposedField MFFloat    groundAngle []
exposedField MFColor    groundColor []
exposedField MFString    backUrl    []
exposedField MFString    bottomUrl   []
exposedField MFString    frontUrl    []
exposedField MFString    leftUrl     []
exposedField MFString    rightUrl    []
exposedField MFString    topUrl      []
exposedField MFFloat    skyAngle     []
exposedField MFColor    skyColor     [ 0 0 0 ]
exposedField MFBool     isBound
}

```

Background Nodes geben einen Hintergrund an, der den Himmel, Boden oder ein Panorama darstellen kann.

3.6.6.5 Billboard

```

Billboard
{
    eventIn      MFNode    addChildren
    eventIn      MFNode    removeChildren
    exposedField SFVec3f    axisOfRotation 0 1 0
    exposedField MFNode    children        []
    field        SFVec3f    bboxCenter     0 0 0
    field        SFVec3f    bboxSize       -1 -1 -1
}

```

Billboard ist eine Gruppierungsnode die ihr lokales Koordinatensystem so abändert, daß die Z-Achse auf den Betrachter zeigt.

3.6.6.6 Box

```

Box
{
    field SFVec3f size 2 2 2
}

```

Die Box Node zeichnet einen Quader.

3.6.6.7 Collision

```
Collision
{
  eventIn      MFNode   addChildren
  eventIn      MFNode   removeChildren
  exposedField MFNode   children      []
  exposedField SFBool   collide       TRUE
  field        SFVec3f  bboxCenter    0 0 0
  field        SFVec3f  bboxSize      -1 -1 -1
  field        SFNode   proxy NULL
  eventOut     SFTIME   collideTime
}
```

Standardmäßig werden alle Objekte auf Kollision mit dem Benutzer (Avatar) getestet. Die Collision-Node bietet die Möglichkeit, die Kollisionsprüfung für ihre Childnodes abzuschalten oder zu modifizieren.

3.6.6.8 Color

```
Color
{
  exposedField MFCOLOR color []
}
```

Diese Node ermöglicht es RGB Farben zu definieren, die dann in Feldern anderer Nodes benutzt werden können.

3.6.6.9 ColorInterpolator

```
ColorInterpolator
{
  eventIn      SFFloat  set_fraction
  exposedField MFFloat  key          []
  exposedField MFCOLOR  keyValue     []
  eventOut     SFCOLOR  value_changed
}
```

Diese Node interpoliert zwischen einer Menge von Farben und sendet jeden neuen Wert als Event (*value_changed*).

3.6.6.10 Cone

```
Cone
{
  field SFFloat  bottomRadius 1
  field SFFloat  height       2
  field SFBool   side         TRUE
  field SFBool   bottom       TRUE
}
```

Diese Node stellt einen Kegel dar.

3.6.6.11 Coordinate

```
Coordinate
{
```

```

    exposedField MFVec3f point []
}

```

Definiert eine Menge von 3D Vektoren, die in anderen Nodes (z.B. IndexedFaceSet) verwendet werden können.

3.6.6.12 CoordinateInterpolator

```

CoordinateInterpolator
{
    eventIn      SFFloat  set_fraction
    exposedField MFFloat  key          []
    exposedField MFVec3f  keyValue     []
    eventOut     MFVec3f  value_changed
}

```

Diese Node interpoliert linear zwischen einer Menge von 3D Vektoren und sendet die berechneten Werte als Event.

3.6.6.13 Cylinder

```

Cylinder
{
    field SFBool  bottom  TRUE
    field SFFloat height  2
    field SFFloat radius  1
    field SFBool  side    TRUE
    field SFBool  top     TRUE
}

```

Erzeugt einen Zylinder.

3.6.6.14 CylinderSensor

```

CylinderSensor
{
    exposedField SFBool    autoOffset      TRUE
    exposedField SFFloat   diskAngle       0.262
    exposedField SFBool    enabled         TRUE
    exposedField SFFloat   maxAngle        -1
    exposedField SFFloat   minAngle        0
    exposedField SFFloat   offset          0
    eventOut     SFBool    isActive
    eventOut     SFRotation rotation_changed
    eventOut     SFVec3f   trackPoint_changed
}

```

CylinderSensor bildet die Mausbewegung auf eine Rotation eines unsichtbaren Zylinders ab, der an der lokalen Y-Achse ausgerichtet ist.

3.6.6.15 DirectionalLight

```

DirectionalLight
{
    exposedField SFFloat  ambientIntensity 0
    exposedField SFColor  color             1 1 1
}

```

```

    exposedField SFVec3f direction      0 0 -1
    exposedField SFFloat intensity      1
    exposedField SFBool  on             TRUE
}

```

DirectionalLight stellt eine Lichtquelle dar, deren Strahlen parallel zu einem angegebenen dreidimensionalen Vektor verlaufen.

3.6.6.16 ElevationGrid

```

ElevationGrid
{
    eventIn      MFFloat  set_height
    exposedField SFNode   color           NULL
    exposedField SFNode   normal          NULL
    exposedField SFNode   texCoord        NULL
    field        MFFloat  height          []
    field        SFBool   ccw             TRUE
    field        SFBool   colorPerVertex  TRUE
    field        SFFloat  creaseAngle     0
    field        SFBool   normalPerVertex TRUE
    field        SFBool   solid           TRUE
    field        SFInt32  xDimension      0
    field        SFFloat  xSpacing        0.0
    field        SFInt32  zDimension      0
    field        SFFloat  zSpacing        0.0
}

```

ElevationGrid definiert ein rechteckiges Gitter mit variabler Höhe in der XZ-Ebene des lokalen Koordinatensystems.

3.6.6.17 Extrusion

```

Extrusion
{
    eventIn MFVec2f    set_crosssubsection
    eventIn MFRotation set_orientation
    eventIn MFVec2f    set_scale
    eventIn MFVec3f    set_spine
    field  SFBool      beginCap          TRUE
    field  SFBool      ccw                TRUE
    field  SFBool      convex            TRUE
    field  SFFloat      creaseAngle       0
    field  MFVec2f      crosssubsection   [ 1 1, 1 -1, -1 -1, -1 1, 1 1 ]
    field  SFBool      endCap            TRUE
    field  MFRotation   orientation       0 0 1 0
    field  MFVec2f      scale             1 1
    field  SFBool      solid             TRUE
    field  MFVec3f      spine            [ 0 0 0, 0 1 0 ]
}

```

Die Extrusion Node definiert geometrische Körper durch die Verschiebung eines zweidimensionalen Objekts entlang einer Raumkurve. Das zweidimensionale Objekt kann an jedem Punkt der Raumkurve skaliert und rotiert werden.

3.6.6.18 Fog

```

Fog
{
    exposedField SFColor   color           1 1 1
    exposedField SFString  fogType        "LINEAR"
    exposedField SFFloat   visibilityRange 0
    eventIn      SFBool    set_bind
    eventOut     SFBool    isBound
}

```

Die Fog-Node bietet die Möglichkeit atmosphärische Effekte (z.B. Nebel) zu simulieren.

3.6.6.19 FontStyle

```

FontStyle
{
    field SFString  family      "SERIF"
    field SFBool    horizontal  TRUE
    field MFString  justify     "BEGIN"
    field SFString  language    ""
    field SFBool    leftToRight TRUE
    field SFFloat   size        1.0
    field SFFloat   spacing     1.0
    field SFString  style       "PLAIN"
    field SFBool    topToBottom TRUE
}

```

Erlaubt es verschiedene Fonts zu benutzen.

3.6.6.20 Group

```

Group
{
    eventIn      MFNode  addChildren
    eventIn      MFNode  removeChildren
    exposedField MFNode  children    []
    field        SFVec3f bboxCenter  0 0 0
    field        SFVec3f bboxSize    -1 -1 -1
}

```

Gruppiert Nodes.

3.6.6.21 ImageTexture

```

ImageTexture
{
    exposedField MFString url        []
    field        SFBool  repeatS    TRUE
    field        SFBool  repeatT    TRUE
}

```

ImageTexture definiert eine Textur durch Angabe eines Bildes und Parametern, die die Abbildung der Textur auf Objekte beschreiben.

3.6.6.22 IndexedFaceSet

```
IndexedFaceSet
{
  eventIn      MFInt32  set_colorIndex
  eventIn      MFInt32  set_coordIndex
  eventIn      MFInt32  set_normalIndex
  eventIn      MFInt32  set_texCoordIndex
  exposedField SFNode   color          NULL
  exposedField SFNode   coord          NULL
  exposedField SFNode   normal         NULL
  exposedField SFNode   texCoord       NULL
  field        SFBool   ccw            TRUE
  field        MFInt32  colorIndex     []
  field        SFBool   colorPerVertex TRUE
  field        SFBool   convex         TRUE
  field        MFInt32  coordIndex     []
  field        SFFloat  creaseAngle    0
  field        MFInt32  normalIndex    []
  field        SFBool   normalPerVertex TRUE
  field        SFBool   solid          TRUE
  field        MFInt32  texCoordIndex  []
}
```

Erlaubt es, dreidimensionale Körper durch die Verwendung von Polygonen zu definieren.

3.6.6.23 IndexedLineSet

```
IndexedLineSet {
eventIn MFInt32 set_colorIndex
eventIn MFInt32 set_coordIndex
exposedField SFNode color NULL
exposedField SFNode coord NULL
field MFInt32 colorIndex []
field SFBool colorPerVertex TRUE
field MFInt32 coordIndex []
}
```

Erlaubt es, dreidimensionale Körper durch die Verwendung von Polylinien zu definieren.

3.6.6.24 Inline

```
Inline
{
  exposedField MFString url      []
  field        SFVec3f  bboxCenter 0 0 0
  field        SFVec3f  bboxSize  -1 -1 -1
}
```

Inline ist eine Gruppierungsnode, die seine Kinder aus dem World Wide Web liest.

3.6.6.25 LOD

```
LOD
{
  exposedField MFNode  level  []
}
```



```

    field      SFVec3f  center  0 0 0
    field      MFFloat  range   []
}

```

Die LOD-Node spezifiziert verschiedene Detaillierungsstufen eines Objekts, die je nach Abstand des Betrachters vom Objekt benutzt werden.

3.6.6.26 Material

```

Material
{
    exposedField SFFloat  ambientIntensity  0.2
    exposedField SFColor  diffuseColor      0.8 0.8 0.8
    exposedField SFColor  emissiveColor      0 0 0
    exposedField SFFloat  shininess          0.2
    exposedField SFColor  specularColor      0 0 0
    exposedField SFFloat  transparency        0
}

```

Die Material-Node definiert die Oberflächeneigenschaften eines Materials

3.6.6.27 MovieTexture

```

MovieTexture
{
    exposedField SFBool    loop              FALSE
    exposedField SFFloat   speed              1
    exposedField SFTIME    startTime          0
    exposedField SFTIME    stopTime           0
    exposedField MFString  url                []
    field        SFBool    repeatS            TRUE
    field        SFBool    repeatT            TRUE
    eventOut     SFFloat   duration_changed
    eventOut     SFBool    isActive
}

```

Definiert eine zeitabhängige Textur, die sich in einer Filmdatei befindet.

3.6.6.28 NavigationInfo

```

NavigationInfo
{
    eventIn      SFBool    set_bind
    exposedField MFFloat   avatarSize        [ 0.25, 1.6, 0.75 ]
    exposedField SFBool    headlight          TRUE
    exposedField SFFloat   speed              1.0
    exposedField MFString  type               "WALK"
    exposedField SFFloat   visibilityLimit    0.0
    eventOut     SFBool    isBound
}

```

NavigationInfo enthält Informationen, die bestimmte Eigenschaften des Anzeigeprogramms beeinflussen.

3.6.6.29 Normal

Normal

```
{
  exposedField MFVec3f  vector  []
}
```

Erlaubt es, eine Menge von 3D Normalenvektoren zu definieren, die in anderen Nodes verwendet werden können.

3.6.6.30 NormalInterpolator

NormalInterpolator

```
{
  eventIn      SFFloat  set_fraction
  exposedField MFFloat  key          []
  exposedField MFVec3f  keyValue     []
  eventOut     MFVec3f  value_changed
}
```

Interpoliert zwischen einer Menge von 3D Vektoren mit der Eigenschaft, daß alle Ausgabevektoren normalisiert sind.

3.6.6.31 OrientationInterpolator

OrientationInterpolator

```
{
  eventIn      SFFloat      set_fraction
  exposedField MFFloat      key          []
  exposedField MFRotation   keyValue     []
  eventOut     SFRotation   value_changed
}
```

Diese Node interpoliert zwischen einer Menge von SFRotation Werten.

3.6.6.32 PixelTexture

PixelTexture

```
{
  exposedField SFImage  image  0 0 0
  field SFBool  repeatS  TRUE
  field SFBool  repeatT  TRUE
}
```

Definiert eine zweidimensionale Textur durch explizit angegebene Pixel.

3.6.6.33 PlaneSensor

PlaneSensor

```
{
  exposedField SFBool  autoOffset      TRUE
  exposedField SFBool  enabled          TRUE
  exposedField SFVec2f  maxPosition     -1 -1
  exposedField SFVec2f  minPosition     0 0
  exposedField SFVec3f  offset          0 0 0
  eventOut     SFBool  isActive
}
```

```

    eventOut      SFVec3f  trackPoint_changed
    eventOut      SFVec3f  translation_changed
}

```

Der PlaneSensor bildet die Bewegungen des Zeigegeräts (z.B. Maus) auf die XY-Ebene seines lokalen Koordinatensystems ab und benutzt die nachfolgende Geometrie seiner *Parent Node* um Treffer zu berechnen.

3.6.6.34 PointLight

```

PointLight
{
    exposedField  SFFloat  ambientIntensity  0
    exposedField  SFVec3f  attenuation        1 0 0
    exposedField  SFCOLOR  color              1 1 1
    exposedField  SFFloat  intensity          1
    exposedField  SFVec3f  location            0 0 0
    exposedField  SFBool   on                  TRUE
    exposedField  SFFloat  radius              100
}

```

Erzeugt ein Punktlicht im Raum. Die Lichtstrahlen werden in alle Richtungen ausgestrahlt.

3.6.6.35 PointSet

```

PointSet
{
    exposedField  SFNode   color  NULL
    exposedField  SFNode   coord  NULL
}

```

Gibt eine Menge dreidimensionaler Punkte mit zugehöriger Farbe an.

3.6.6.36 PositionInterpolator

```

PositionInterpolator
{
    eventIn      SFFloat  set_fraction
    exposedField  MFFloat  key          []
    exposedField  MFVec3f  keyValue     []
    eventOut      SFVec3f  value_changed
}

```

Interpoliert linear zwischen SFVec3f Werten.

3.6.6.37 ProximitySensor

```

ProximitySensor
{
    exposedField  SFVec3f  center          0 0 0
    exposedField  SFVec3f  size            0 0 0
    exposedField  SFBool   enabled         TRUE
    eventOut      SFBool   isActive
    eventOut      SFVec3f  position_changed
    eventOut      SFRotation orientation_changed
    eventOut      SFTIME   enterTime
}

```

```

    eventOut      STime      exitTime
}

```

Dieser Sensor gibt Events aus, wenn der Benutzer sich an einem bestimmten Raumabschnitt (quaderförmig) befindet.

3.6.6.38 ScalarInterpolator

```

ScalarInterpolator
{
    eventIn      SFFloat  set_fraction
    exposedField MFFloat  key          []
    exposedField MFFloat  keyValue     []
    eventOut      SFFloat  value_changed
}

```

Interpoliert zwischen einer Menge von SFFloat Werten.

3.6.6.39 Script

```

Script
{
    exposedField MFString      url          []
    field        SFBool        directOutput FALSE
    field        SFBool        mustEvaluate FALSE
    # Eine beliebige Anzahl von:
    eventIn      eventName     eventName
    field        fieldName     fieldName     initialValue
    eventOut      eventName     eventName
}

```

Erlaubt es, Programmiersprachen einzubinden.

3.6.6.40 Shape

```

Shape
{
    exposedField SFNode  appearance  NULL
    exposedField SFNode  geometry    NULL
}

```

Die Shape Node erlaubt es, *gerenderte* Objekte zu erzeugen. Das Aussehen des Objekts wird durch *appearance* und die Form durch *geometry* bestimmt.

3.6.6.41 Sound

```

Sound
{
    exposedField SFVec3f  direction  0 0 1
    exposedField SFFloat  intensity  1
    exposedField SFVec3f  location   0 0 0
    exposedField SFFloat  maxBack    10
    exposedField SFFloat  maxFront   10
    exposedField SFFloat  minBack    1
    exposedField SFFloat  minFront   1
    exposedField SFFloat  priority    0
}

```

```

    exposedField SFNode    source      NULL
    field        SFBool    spatialize  TRUE
}

```

Sound beschreibt die Position und spezielle Präsentation eines Klanges in einer VRML Szene.

3.6.6.42 Sphere

```

Sphere
{
    field SFFloat radius 1
}

```

Beschreibt eine Kugel.

3.6.6.43 SphereSensor

```

SphereSensor
{
    exposedField SFBool    autoOffset      TRUE
    exposedField SFBool    enabled         TRUE
    exposedField SFRotation offset         0 1 0 0
    eventOut     SFBool    isActive
    eventOut     SFRotation rotation_changed
    eventOut     SFVec3f   trackPoint_changed
}

```

SphereSensor bildet die Mausbewegung auf eine Rotation einer unsichtbaren Kugel um den lokalen Ursprung.

3.6.6.44 SpotLight

```

SpotLight
{
    exposedField SFFloat ambientIntensity 0
    exposedField SFVec3f attenuation      1 0 0
    exposedField SFFloat beamWidth        1.570796
    exposedField SFColor color             1 1 1
    exposedField SFFloat cutOffAngle       0.785398
    exposedField SFVec3f direction         0 0 -1
    exposedField SFFloat intensity         1
    exposedField SFVec3f location          0 0 0
    exposedField SFBool  on                TRUE
    exposedField SFFloat radius            100
}

```

Eine Lichtquelle, deren Lichtstrahlen kegelförmig von einem bestimmten Punkt ausgehen.

3.6.6.45 Switch

```

Switch
{
    exposedField MFNode    choice          []
    exposedField SFInt32   whichChoice     -1
}

```

Diese Gruppierungsnode erlaubt es, eine einzelne Kindnode zu aktivieren. Die Nichtaktivierten Kindnodes empfangen jedoch weiterhin Events.

3.6.6.46 Text

```
Text {
  exposedField MFString string []
  exposedField SFNode fontStyle NULL
  exposedField MFFloat length []
  exposedField SFFloat maxExtent 0.0
}
```

Gibt zweidimensionalen Text im dreidimensionalen Raum aus.

3.6.6.47 TextureCoordinate

```
TextureCoordinate
{
  exposedField MFVec2f point []
}
```

Definiert eine Menge von 2D Texturkoordinaten, die von Vertex-basierten Geometrienodes benutzt werden.

3.6.6.48 TextureTransform

```
TextureTransform
{
  exposedField SFVec2f center 0 0
  exposedField SFFloat rotation 0
  exposedField SFVec2f scale 1 1
  exposedField SFVec2f translation 0 0
}
```

Definiert eine 2D Transformation, die einer Texturekoordinaten zugewiesen sind.

3.6.6.49 TimeSensor

```
TimeSensor
{
  exposedField SFTime cycleInterval 1
  exposedField SFBool enabled TRUE
  exposedField SFBool loop FALSE
  exposedField SFTime startTime 0
  exposedField SFTime stopTime 0
  eventOut SFTime cycleTime
  eventOut SFFloat fraction_changed
  eventOut SFBool isActive
  eventOut SFTime time
}
```

Die TimeSensor Node sendet Events in bestimmten Zeitintervallen aus.

3.6.6.50 TouchSensor

```
TouchSensor
{
  exposedField SFBool enabled TRUE
  eventOut SFVec3f hitNormal_changed
}
```

```

    eventOut    SFVec3f    hitPoint_changed
    eventOut    SFVec2f    hitTexCoord_changed
    eventOut    SFBool     isActive
    eventOut    SFBool     isOver
    eventOut    SFTIME     touchTime
}

```

Sendet Events wenn der Benutzer auf Teile der Geometrie klickt, die zur Elterngruppe des TouchSensors gehören.

3.6.6.51 Transform

```

Transform
{
    eventIn      MFNode     addChildren
    eventIn      MFNode     removeChildren
    exposedField SFVec3f     center          0 0 0
    exposedField MFNode     children          []
    exposedField SFRotation  rotation         0 0 1 0
    exposedField SFVec3f     scale            1 1 1
    exposedField SFRotation  scaleOrientation 0 0 1 0
    exposedField SFVec3f     translation      0 0 0
    field        SFVec3f     bboxCenter       0 0 0
    field        SFVec3f     bboxSize         -1 -1 -1
}

```

Definiert ein Koordinatensystem für seine Kinder, das relativ zum Koordinatensystem seiner Eltern ist.

3.6.6.52 Viewpoint

```

Viewpoint
{
    eventIn      SFBool     set_bind
    exposedField SFFloat     fieldOfView     0.785398
    exposedField SFBool     jump             TRUE
    exposedField SFRotation  orientation     0 0 1 0
    exposedField SFVec3f     position        0 0 10
    field        SFString    description     ""
    eventOut     SFTIME     bindTime
    eventOut     SFBool     isBound
}

```

Definiert eine bestimmte Position in einem lokalen Koordinatensystem, von dem der Benutzer die Szene betrachten kann.

3.6.6.53 VisibilitySensor

```

VisibilitySensor
{
    exposedField SFVec3f     center          0 0 0
    exposedField SFBool     enabled          TRUE
    exposedField SFVec3f     size            0 0 0
    eventOut     SFTIME     enterTime
    eventOut     SFTIME     exitTime
    eventOut     SFBool     isActive
}

```

Die VisibilitySensor Node stellt fest, wenn ein quaderförmiges Gebiet nicht mehr sichtbar ist.

3.6.6.54 WorldInfo

```
WorldInfo
{
  field MFString  info    []
  field SFString  title   ""
}
```

Enthält Informationen über die Welt. Sie dient ausschließlich zu Dokumentationszwecken.

3.6.7 Erweiterbarkeit (Prototypen)

Die Erweiterbarkeit von VRML wird möglich durch das sogenannte *Prototyping*. Es erlaubt die Definition von neuen Nodes. Eine Prototypdefinition ist wie folgt aufgebaut:

- Das **PROTO** Schlüsselwort
- Der **Name** des neuen Nodetyps
- Die **Prototypdeklaration** die aus folgenden Teilen besteht:
 - Eine Liste der öffentlichen eventIns und eventOuts
 - Eine Liste der öffentlichen *exposedFields* und Feldern mit ihren Standardwerten

Die Event- bzw. Felderliste wird von eckigen Klammern eingeschlossen

- Die **Prototypdefinition**, die aus einer oder mehreren Nodes, Prototypen und Routes besteht. Diese verwendeten Nodes können das *IS* Schlüsselwort enthalten. Dieses assoziiert Felder und Eventnamen, die in der Prototypendeklaration enthalten sind, mit Feldern und Events in der Prototypdefinition.

Die Syntax ist folgendermaßen:

```
PROTO prototypname [ eventIn      eventtypname name
                     eventOut     eventtypname name
                     exposedField feldtypname  name standardwert
                     field        feldtypname  name standardwert
                     ...
                     ]
{
  Mehrere oder keine Routes und Prototypen
  Erste Node (Definiert den Nodetyp des Prototyps)
  Mehrere oder keine Nodes (jedes Typs), Routes und Prototypen
}
```

Hierzu ein Beispiel:

```
#VRML V2.0 utf8

#####
# Hier kommt der Prototyp
#####

PROTO SphereCone [ field SFFloat radius    2.0
                   field SFFloat height   5.0
```



```

        field SFNode  sphereApp NULL
        field SFNode  coneApp   NULL ]
{
  Transform
  {
    children [ Shape
      {
        appearance IS sphereApp
        geometry Sphere { radius IS radius }
      }
      Shape
      {
        appearance IS coneApp
        geometry Cone { height IS height }
      }
    ]
  }
}

#####
# ... und ab hier wird er verwendet
#####

Transform
{
  translation 15 0 0
  children SphereCone
  {
    radius 5.0
    height 20.0
    sphereApp Appearance { material Material { ... } }
    coneApp   Appearance { texture ImageTexture { ... } }
  }
}

Transform
{
  translation -10 0 0
  children SphereCone
  { # Hier werden die Standardwerte benutzt
    sphereApp Appearance { texture ImageTexture { ... } }
    coneApp   Appearance { material Material { ... } }
  }
}

```

Es ist auch möglich auf Prototypen in anderen Dateien zuzugreifen

```

EXTERNPROTO prototypname [ eventIn      eventtypname name
                           eventOut     eventtypname name
                           exposedField feldtypname  name
                           feldtypname  feldtypname  name
                           ...
                           ]
"URL/URN" (, "URL/URN", "URL/URN", ... )

```

Der Prototyp wird dann in der Datei gesucht, auf die durch die URL oder URN verwiesen wird. Werden mehrere URL's oder URN's angegeben, werden sie der Reihenfolge nach durchsucht.

3.6.8 Verwendung von Scripts

Oft werden Entscheidungslogiken und Funktionalitäten benötigt, die nur durch die vorhandenen Nodes nicht realisierbar sind¹⁰. Für diesen Zweck gibt es die *Scriptnode*. Eine Scriptnode wird aktiviert, wenn ein bestimmter Event sie erreicht. Sie kann dann Aktionen ausführen wie das Versenden von Events, Berechnungen durchführen, mit Servern im Internet kommunizieren, usw.

Die Scriptnode verweist auf eine URL, an der sich ein Programm oder ein Skript befindet. Dieses kann in jeder Sprache geschrieben sein, die der Browser versteht.

Hierzu ein Beispiel: Dieses Script merkt sich den Zustand einer Tür (offen/ geschlossen) und ob die nötige Sicherheitskombination, um die Tür zu öffnen, eingegeben wurde. Den Status der Tür sendet das Script in bestimmten Zeitintervallen.

```
DEF OpenVault Script
{
  eventIn  STime  openVault
  eventIn  SFBool  combinationEntered
  eventOut STime  vaultUnlocked
  field    SFBool  unlocked FALSE

  url "javascript:
      function combinationEntered(value)
      {
        unlocked = value;
      }

      function openVault(value)
      {
        if (unlocked)
          vaultUnlocked = value;
      }"
}
```

Um genaue Details der Anbindung an Java und Javascript zu erfahren, siehe [GB96] Appendix C und D.

3.7 Anwendungen von VRML

Zum Abschluß werden einige Beispiele für den Einsatz von VRML gegeben. Einige Einsatzgebiete lassen sich erkennen und grob untergliedern:

- Architektur
- Chemie
- Cyberworld (Chat)
- Mathematik
- Online-Shopping
- Virtuelle Messen und

¹⁰Bsp.: Wenn die Tür geschlossen ist UND die richtige Zahlenkombination eingegeben wurde DANN öffne die Tür

3.7.1 Architektur

Zur Visualisierung von Architekturen nutzt ein deutscher Fertighausanbieter¹¹ VRML. Der Benutzer kann einen virtuellen Rundgang durch die angebotenen Häuser machen. Er kann sich auf diese Weise schon einen Eindruck der Räumlichkeiten des Hauses machen.

3.7.2 Chemie

In der Chemie treten immer komplexere Moleküle auf, die schwer vorstellbar sind. Hierbei ist die dreidimensionale Darstellung von Molekülen eine unentbehrliche Hilfe.

Auch die Darstellung von chemischen und biologischen Abläufen kann dargestellt werden. Viele Beispiele findet man unter <http://www.pc.chemie.th-darmstadt.de/vrml/>.

3.7.3 Virtuelle Welten

In virtuellen Welten können sich Menschen treffen die durch Avatare (s.o.) repräsentiert werden. Das Projekt *Worlds*¹² stellt solche Welten bereit. Diese können von den Benutzern erweitert werden¹³ und den eigenen Anforderungen angepasst werden.

Es werden auch schon virtuelle Messen abgehalten, in denen Firmen ihre Produkte vorstellen. Wie auf einer echten Messe können die Benutzer Firmenstände aufsuchen und die Produkte begutachten. Auch die Kommunikation mit anderen Messebesuchern ist möglich.

3.7.4 Mathematik

In der Mathematik wird VRML zur Anzeige von dreidimensionalen mathematischen Objekten eingesetzt.

Einen ganzen „Zoo“ dieser Objekte findet man auf der Seite <http://www.geom.umn.edu:80/software/weboogl/zoo/>.

3.8 Literatur

- [GB95] Mark Pesce Gavin Bell, Anthony Parisi. The Virtual Reality Modeling Language: Version 1.0 Specification. URL: <http://vrml.wired.com/vrml.tech/vrmlspec.html>, 5 1995.
- [GB96] Chris Marrin Gavin Bell, Rikk Carey. The Virtual Reality Modeling Language: Version 2.0 Specification. URL: <http://vrml.sgi.com/moving-worlds/index.html>, 8 1996.
- [TS95] Florian Wenz Tom Sperlich. Cyber-Web: Dreidimensionale Welten auf dem Internet. *c't*, 7, 1995.
- [TS96] Florian Wenz Tom Sperlich. Welten im Netz: VRML 2.0: erweiterter 3D-Standard fürs Internet. *c't*, 8, 1996.

¹¹Siehe <http://www.kampa.de/vrml.html>

¹²<http://www.kaworlds.com/>

¹³Es gibt eine vorgegebene Hauptstraße, an diese können dann neue Straßen angeschlossen werden, neue Häuser können erbaut werden, usw.



Kapitel 4

Java und Java Applets

Günther Augustin

4.1 Übersicht

Dieser Beitrag gibt einen Überblick über die Programmiersprache Java und die Java-Programme, die als Java-Applets bezeichnet werden (Java Applets sind im Vergleich zu normalen Anwendungsprogrammen in Java eingeschränkt, bezüglich der möglichen Benutzerinteraktionen, der Dateisystem- und Netzzugriffe sowie dem Lesen und Manipulieren von Systemparametern). Ferner wird erläutert, wie Java-Applets in WWW-Seiten eingebettet und ausgeführt werden.

4.2 Einleitung

Das World Wide Web (WWW) erfreut sich in den letzten Monaten und Jahren einer explosionsartig wachsenden Beliebtheit. Die Bandbreite des Internet hinkt dieser steigenden Nutzung stark hinterher. Das Netz ist somit regelmäßig überlastet. Ein Ansatz, um diese Überlastung abzumildern, ist die Idee, statt eines erheblichen Datenvolumens jeweils ein geringes Codevolumen, das inzwischen als Applet bezeichnet wird, zu übertragen, wobei dieses Programm dann seinerseits die entsprechenden Daten vor Ort produzieren kann. Darüber hinaus kommt durch die Übermittlung von Programmen natürlich noch eine zusätzliche vielseitig nutzbare Funktionalität hinzu. Die von der Firma Sun Microsystems ab etwa 1993 ursprünglich für den Einsatz in intelligenten Haushaltsgeräten vorgesehene Programmiersprache Java erwies sich für derartige Zwecke als geeignet. Im Folgenden wird die Programmiersprache Java und die Einbettung von Java-Applets in das WWW aufbauend auf [Fla96] erläutert. Ferner werden Sicherheitsaspekte beim Einsatz von Applets besprochen.

4.3 Spracheigenschaften

Java ist eine allgemeine Programmiersprache. Java-Programme werden in einen plattformunabhängigen Bytecode übersetzt und danach ausgeführt (siehe 4.4).

4.3.1 Primitive Datentypen

Java bietet die primitiven Datentypen: **boolean**, **char** sowie die numerischen Datentypen **byte**, **short**, **int**, **long**, **float**, **double**. Bemerkenswert ist dabei die Implementierung von **char** in Unicode-Codierung (16 bit) womit die meisten auf der Welt bekannten Alphabete und Schriften in einem Code darstellbar sind. Zu den numerischen Datentypen ist anzumerken, daß alle vorzeichenbehaftet sind, für alle die Genauigkeit (Zahl der bits in der internen Darstellung) vom Sprachstandard plattformunabhängig vorgegeben ist. Auch die Rechenoperationen für diese Datentypen sind gemäß IEEE 754 festgelegt. Als weitere Datentypen sind in Java **String** und **StringBuffer** vordefiniert. **String** sind dabei nur einmal zuweisbar, während **StringBuffer** ihren Inhalt zur Laufzeit wiederholt ändern dürfen.

4.3.2 Höhere Datentypen

Java bietet den Datentyp „Zeiger“ nicht an. Als Konzept sind lediglich Referenzen auf Objekte definiert, und diese sind auch nur als solche zu benutzen. Insbesondere ist es nicht möglich, mit Zeigern zu rechnen oder Referenzen auf nicht mehr existente oder nicht zugreifbare Objekte zu erzeugen. Durch diese Einschränkung ist bei der Programmierung in Java eine Reihe von Fehlern ausgeschlossen, die bei der Programmierung in anderen Sprachen erhebliche Probleme bereiten. Als zusammengesetzter Datentyp werden außer Objekten (siehe 4.3.5) Arrays implementiert. Diese werden in Java automatisch initialisiert (jede Variable ist in Java letztlich einer Klasse zugeordnet und jede Klasse hat einen Standard-Konstruktor, also kann auch jedes Feld eines Arrays automatisch initialisiert werden). Die Indizes von Arrayzugriffen werden zur Laufzeit überprüft. Unzulässige Zugriffe verursachen **Exceptions** (siehe 4.3.4).

4.3.3 Speicherverwaltung

Die Zuordnung und Freigabe der Ressource Speicherplatz erfolgt in Java automatisch. Beim Erzeugen eines Objekts wird ihm der benötigte Speicherplatz zur Verfügung gestellt (alle Objekte stammen von der Klasse **Object** ab, deren Konstruktor als einziger tatsächlich vom System Speicher anfordern kann). Die Freigabe nicht mehr benötigten Speichers (der durch nicht mehr zugreifbare Objekte belegt ist) erfolgt mittels Garbage Collection.

4.3.4 Fehler- und Ausnahmebehandlung

Java bietet **Exceptions**, die eine elegante und effektive Fehler- und Ausnahmebehandlung ermöglichen. Das dazu verwendete Sprachkonstrukt sieht wie folgt aus:

```
try { Operation }
catch ( Exception1 ) { Behandlung1 }
catch ( Exception2 ) { Behandlung2 }
...
finally { Aufräumen }
```

Eine kritische Operation wird ausgeführt. Wenn dabei ein Fehler auftritt, wird eine **Exception** ausgelöst (im Java-Kontext spricht man davon daß eine **Exception** „geworfen“ wird). Zum Beispiel verursacht ein unzulässiger Array-Zugriff eine **ArrayIndexOutOfBoundsException** während ein Zugriff auf eine nicht vorhandene Datei eine **FileNotFoundException** „wirft“. Eine Division durch 0 hat eine **ArithmeticException** zur Folge. Falls eine **Exception** auftritt wird die Operation abgebrochen und das jeweilige **catch**-Konstrukt „fängt“ die **Exception** auf und veranlaßt die dafür vorgesehene Behandlung. In jedem Falle (ob nun eine **Exception** auftritt oder nicht) wird anschließend der durch **finally** definierte Block ausgeführt um eine Nachbereitung der Operation zu gewährleisten (z.B. Schließen von Dateien etc.).

4.3.5 Objektorientiertes Programmieren

Java folgt dem Paradigma des Objektorientierten Programmierens (mit Klassendefinition, Kapselung, Vererbung, Instatiation etc.). Es ermöglicht die Kapselung von Methoden und Attributen in Klassen, die Instantiierung von Klassen in Form von Objekten sowie Vererbung unter Klassen. Ferner ist das Konzept des method overriding realisiert, d.h. daß eine Unterklasse die geerbten Methoden neu definieren und damit überschreiben kann. Weiterhin ist das Konzept des method overloading umgesetzt worden, d.h. daß in einer Klasse mehrere Methoden mit gleichem Namen aber unterschiedlicher Signatur existieren können (wobei allerdings nur die Parameterliste berücksichtigt wird und der Ergebnistyp der Methode keine Rolle spielen darf). Entsprechend der Parameterliste beim Methodenaufruf wird die jeweils passende Methode benutzt. Außer Objekten gibt es in Java keine globalen Variablen, lediglich Klassen-Variablen und -Methoden. Klassen-Variablen sind für jede Klasse nur einmal vorhanden und

nicht einem bestimmten Objekt sondern einer Klasse zugeordnet (eine Klassen-Variable könnte z.B. die Zahl der vorhandenen Objekte einer Klasse aufnehmen). Klassen-Methoden sind Methoden die nur auf Klassen-Variablen operieren d.h. ihre Ausführung ist nicht an das Vorhandensein einer Instanz dieser Klasse gebunden. Java bietet des weiteren das Konzept der abstrakten Klassen. Dies sind Klassen, bei denen für mindestens eine Methode zwar die Signatur, aber nicht die Implementierung der jeweiligen (abstrakten) Methode feststeht. Dies hat zur Folge daß man keine Objekte dieser Klasse instantiiieren kann. Eine andere Klasse kann aber von dieser abstrakten Klasse abgeleitet werden und ihrerseits eine Implementation der abstrakten Methode definieren. Von einer abstrakten Klasse kann aber auch wieder eine abstrakte Klasse abgeleitet werden (wobei bisher abstrakte Methoden durch Implementierung zu konkreten Methoden werden können, aber nicht umgekehrt - es könnten allerdings neue abstrakte Methoden definiert werden). Mit Vererbung ist in Java stets Einfachvererbung gemeint, d.h. eine Klasse hat stets nur eine einzige Oberklasse (ist also von dieser abgeleitet). Um diese Einschränkung etwas abzumildern, gibt es in Java **Interfaces**. **Interfaces** sind abstrakte Klassen, die keine Attribute haben dürfen und deren sämtliche Methoden abstrakt sein müssen. Mehrfachvererbung ist damit nur insofern möglich, als eine Klasse von einer anderen abgeleitet wird und gleichzeitig ein oder mehrere **Interfaces** implementiert (wohlgemerkt „implementiert“, weil die Klasse nämlich bis auf Klassen-Methoden von einem **Interface** keine Code „erben“ kann).

4.3.6 Packages

Zur Unterstützung der praktischen Programmierarbeit bietet Java standardmäßig mehrere Modulbibliotheken. Diese heißen in Java **Packages**. Der Benutzer kann von sich aus nach Belieben weitere **Packages** erstellen oder von Dritten übernehmen. Zu den wichtigsten **Packages** gehören **java.awt**, **java.io**, **java.net** und **java.lang**.

java.awt implementiert eine objektorientierte graphische Benutzerschnittstelle, die relativ leicht zu benutzen ist und aus Sicht des Java-Programmierers Plattformunabhängigkeit gewährleistet. Die graphische Benutzerschnittstelle besteht auf der für den Java-Programmierer elementarsten Ebene aus einzelnen **Components** (z.B. **Button**, **Checkbox**, **List**, **Scrollbar**). Diese lassen sich in **Containern** (z.B. **Window**, **Panel**, **Dialog**) zusammenfassen, wobei **Container** auch rekursiv ihresgleichen enthalten können. Die Anordnung der einzelnen Elemente eines **Containers** wird von dem diesem **Container** zugeordneten **LayoutManager** bestimmt. Es stehen standardmäßig mehrere **LayoutManager** zur Verfügung, die die Elemente z.B. gitterförmig, wie Fließtext oder als Kartei mit Karteikartenreitern anordnen. Der **LayoutManager** berechnet jeweils aus der Größe der einzelnen Elemente ihre Anordnung und die Größe des gesamten **Containers**. Den einzelnen **Components** und **Container** werden Benutzerinteraktionen (z.B. Tastatureingaben, Mausklicks, Mausbewegungen, Fensteroperationen) über einen speziellen Datentyp **Event** mitgeteilt.

java.io implementiert u.a. Dateisystemzugriffe, IO-Streams, Filter für Streams, Pipelines.

java.net bietet von Kommunikationsroutinen, die auf das WWW und einzelne URLs (Universal Resource Locators) aufbauen, bis hin zu Routinen die über Sockets Kommunikationsverbindungen aufbauen und nutzen, und sogar Routinen mit denen einzelne Datagramme verschickt werden können.

java.lang bietet u.a. ein **Interface** zur Implementierung von **Threads** und **Threadgroups**. Java selbst unterstützt ergänzend dazu die Synchronisation beim Zugriff auf einzelne Datenobjekte bzw. bei der möglichen gleichzeitigen Ausführung von Code-Abschnitten.

4.4 Java-Ausführungsumgebung

Ein fertig erstelltes Java-Programm wird von einem Java-Compiler unter Zuhilfenahme der importierten **Packages** übersetzt. Das Ergebnis ist ein Programm in Java-Bytecode. Für Java-Bytecode ist eine virtuelle plattformunabhängige Maschine (Java Virtual Machine - JVM) definiert. Diese kann nun ihrerseits auf verschiedenen Plattformen implementiert werden und gewährleistet damit, daß jedes Programm in

Java-Bytecode auf der jeweiligen Hardwareplattform ausgeführt werden kann. Inzwischen hat die Firma Sun die Java-Philosophie fortgeschrieben und erstmals Prozessoren gebaut, die direkt Java-Bytecode ausführen können. Diese Prozessoren bilden zusammen mit Java selbst und der Vernetzung über das WWW den Kern einer Systemarchitektur die sich „network computing“ nennt. Dabei wird Rechenleistung und persistente Datenhaltung (z.B. Festplatten) zentral auf Servern konzentriert, während der einzelne Rechner am Arbeitsplatz im Wesentlichen eine moderne graphische Benutzerschnittstelle anbietet und für Zugriffe auf Daten und Programme überwiegend oder gar ausschließlich über das Netz auf den Server zurückgreift.

4.5 Die Schnittstelle zwischen HTML und Java-Applets

Ein Programm in Java-Bytecode kann von einem Programm ausgeführt werden, das die Java Virtual Machine implementiert. Im allgemeinen Falle soll das Java-Bytecode-Programm aber über das Netz übertragen werden, bevor es auf dem Zielrechner (client) ausgeführt wird. Der Zugriff auf das Programm erfolgt durch einen WWW-Browser (der seinerseits die JVM implementiert). In eine WWW-Seite (die in der Sprache HTML - Hypertext Markup Language - verfaßt ist) wird ein Applet eingebettet, wie das bei anderen Elementen auch der Fall ist (z.B. Bilder, Videosequenzen, Audiosequenzen). Sobald der WWW-Browser also in einer eingelesenen Seite diese Einbettung findet, lädt er das entsprechende Java-Bytecode-Programm und führt es aus. Die Einbettung in HTML sieht im einzelnen wie folgt aus:

```

< APPLET
    CODEBASE = applet-url
    CODE = applet-filename
    WIDTH = pixel-width
    HEIGHT = pixel-height
    ALT = alternate-text
    NAME = applet-name
    ALIGN = alignment
    VSPACE = vertical-pixel-space
    HSPACE = horizontal-pixel-space
>
< PARAM NAME = parameter-name1 VALUE = value1 >
< PARAM NAME = parameter-name2 VALUE = value2 >
...
alternate-html
< /APPLET >

```

Die Bedeutung der einzelnen Parameter ergibt sich wie folgt:

applet-url ist die Basis-URL des Applets (falls nicht angegeben ist diese identisch mit dem Verzeichnis in dem die betreffende HTML-Seite steht).

applet-filename ist der Name der Datei, in der der Java-Bytecode für das betreffende Applet steht.

pixel-width und *pixel-height* sind die Dimensionen der Fläche, die dem Applet bei der Darstellung auf dem Bildschirm zur Verfügung gestellt wird.

alternate-text ist der Text, der alternativ von WWW-Browsern dargestellt werden soll, die zwar das `< APPLET >`-tag in HTML erkennen, aber keinen Java-Bytecode ausführen können.

applet-name ist ein evtl. gesondert angegebener Name des Applets, der bei der Kommunikation unter verschiedenen Applets dazu dient, daß diese sich gegenseitig identifizieren.

alignment beschreibt die Anordnung der Applet-Fläche auf der dargestellten HTML-Seite (analog zur Anordnung von Bildern innerhalb eines HTML-Dokuments).

vertical-pixel-space und *horizontal-pixel-space* geben den Umfang des Raums an, der über und unter bzw. rechts und links vom Applet im Kontext der Darstellung auf dem Bildschirm frei bleiben soll.

PARAM NAME ... VALUE ... übergibt jeweils einen Parameter an das Applet, der von diesem zur Laufzeit ausgelesen werden kann. Dies erlaubt die individuelle Parametrisierung des Applets an unterschiedlichen Stellen auf der betreffenden Seite bzw. auf unterschiedlichen Seiten. Das Applet greift dabei auf Methoden der Klasse Applet zurück und erhält für den jeweils angegebenen Parameternamen den aktuellen Wert (jedes Applet ist wie zu erwarten war von der Klasse Applet abgeleitet).

alternate-html wird von WWW-Browser dargestellt, die das `<APPLET>`-tag von HTML nicht kennen.

Für ein gültiges `<APPLET>`-tag sind die Parameter *CODE*, *WIDTH* und *HEIGHT* unbedingt erforderlich, während alle anderen (inkl. *PARAM NAME ... VALUE ...*) optional sind.

Im der Praxis könnte die Übergabe beispielsweise wie folgt realisiert werden:

- Die Einbettung in HTML ruft das Applet **foobar** auf und übergibt an die Parameter **foo** und **bar** die Werte **abc** bzw. **xyz** :

```
<APPLET
  CODE = „foobar.class“
  WIDTH = 300
  HEIGHT = 300
>
<PARAM NAME = foo VALUE = „abc“ >
<PARAM NAME = bar VALUE = „xyz“ >
</APPLET>
```

- Im Java-Applet erfolgt dann das Auslesen der beiden Parameter:

```
protected String foo;
protected String bar;

public void init() {
    foo = new String( this.getParameter(„foo“));
    bar = new String( this.getParameter(„bar“));
    ...
}
```

4.6 Applet-Ausführung

Ein Applet kann wie ein normales Programm seine „Kommandozeilenparameter“ lesen, unterscheidet sich aber durchaus von herkömmlichen Programmen und zwar in der Art und Weise wie es intern mit dem Aktivitätsfaden umgeht. Ein Applet überschreibt einzelne Methoden der Klasse **Applet** und erwartet den Aufruf dieser Methoden zu bestimmten definierten Zeitpunkten. Nach dem Laden des Applets wird einmalig die Methode **init()** aufgerufen. Sobald das Fenster in dem das Applet dargestellt wird sichtbar ist, wird die Methode **start()** aufgerufen, wenn das Fenster nicht mehr sichtbar ist wird **stop()** aufgerufen. Ein Neuzeichnen des Fensters (inkl. Applet) wird durch **paint()** veranlaßt und das endgültige Ende des Ausführung des Applets wird mit **destroy()** signalisiert (hier kann die Freigabe

von Ressourcen und die Sicherung von persistenten Daten des Applets erfolgen). Benutzerinteraktionen mit dem Applet werden über **Events** signalisiert, wobei entsprechende Methoden (z.B. **mouseDown()**, **action()**) aufgerufen werden und ein **Event**-Objekt den Typ der betreffenden Interaktion im Einzelnen beschreibt.

4.7 Applets und Sicherheit

Bei dem bisher geschilderten Vorgehen stellt sich natürlich die Frage, inwiefern dabei die Sicherheit des Systems gefährdet ist. Ein fremdes Programm aus dem Netz zu laden und auszuführen, stellt ein erhebliches Sicherheitsrisiko dar. Java trifft dagegen einige Vorkehrungen. Einerseits gibt es ohne das Konstrukt des Datentyps Zeiger nur eine eingeschränkte Möglichkeit der Manipulation des lokalen Systems. Ferner wird jedes Java-Bytecode-Programm nach dem Laden und vor der Ausführung sowie zur Laufzeit überprüft und erkennbar unzulässige Codefolgen werden erst gar nicht ausgeführt. Ferner kann ein WWW-Browser die erlaubten Zugriffe für ein Applet einschränken und dies passiert auch (bis hin zu dem Ausmaße daß bei manchen Browsern diese Maßnahmen gar nicht gelockert werden können). Die Einschränkungen betreffen u.a. das Lesen und Ändern von Systemparametern, Zugriffe auf das lokale Dateisystem, den Aufruf von anderen Programmen, den Aufbau von Netzverbindungen zu anderen Rechnern (außer dem lokalen Rechner und jenem von dem das Applet geladen wurde). Eingeschränkt ist auch die Identifizierung des aufrufenden Benutzers und das „unbeaufsichtigte“ öffnen von Fenstern (die evtl. andere Programme simulieren könnten). Was jedoch nicht verhindert werden kann ist der Mißbrauch des Systems in Form von übermäßigem Ressourcenverbrauch. All diese Einschränkungen beziehen sich aber nur auf Applets die aus dem Netz geladen wurden. Applets die lokal geladen wurden haben nach wie vor volle „Handlungsfreiheit“. Die neueste Version von Java (Version 1.1) bietet zudem die Möglichkeit Applets zu Zertifizieren indem dem Java-Bytecode eine digitale Unterschrift beigefügt wird und der lokale Nutzer definieren muß, welchen Appletanbietern er „vertraut“.

4.8 Ausblick

Java erweist sich als eine moderne und zukunftsweisende Programmiersprache. Sie weist schon viele Eigenschaften auf die man sich in anderen Programmiersprachen erst durch zusätzliche (vorzugsweise nicht auf allen Plattformen einheitliche) Ergänzungspakete aufbauen muß. Lediglich die bisweilen geringe Ausführungsgeschwindigkeit schlägt negativ zu Buche. Die Architekturform des „network computing“ dürfte sich als ein durchaus aussichtsreiches Modell etablieren, da in Zukunft mit einer steigenden Vernetzung zu rechnen ist. Insbesondere in organisatorisch abgeschlossenen Kontexten (Firmen, Forschungseinrichtungen, Universitäten) stellt der verringerte Wartungsaufwand einen wichtigen Vorteil dar. Die Pflege eines Servers statt einer Vielzahl von Personal Computern ist schneller, ermöglicht Konsistenz und nimmt weniger Ressourcen in Anspruch. Der private Anwender könnte dieses Konzept auch umsetzen, sofern er ein Netz und mehrere Rechnerzugänge mit gleichem oder ähnlichem Nutzungsprofil betreibt. Nicht realistisch erscheint hingegen bei der derzeitigen Kostenstruktur im Kommunikationsbereich die Perspektive nach der sich der einzelne private Anwender bei jeder Programmausführung den jeweiligen Code kostenpflichtig von einem zentralen Server herunterlädt. Dies könnte sich allerdings bei schnelleren und erheblich billigeren Netzen unter Umständen ändern.

4.9 Literatur

[Fla96] David Flanagan. *Java in a Nutshell*. O'Reilly & Associates, Inc., Februar 1996.



Kapitel 5

Electronic Publishing: Ein Überblick

Khaldoun Ateyeh

5.1 Übersicht

Diese Ausarbeitung gibt einen Überblick über die vorhandenen Möglichkeiten zur Entwicklung von Multimedia-Anwendungen und beschreibt einen möglichen Entwicklungsprozeß.

5.2 Einleitung

Multimediale Anwendungsfelder: [Bol94] [Bol95]

- **Unterhaltung**
Der weitaus größte Marktanteil von Multimedia-Applikationen wird im Gebiet der elektronischen Spiele, seien es Abenteuerspiele, Simulationspiele oder Wissensspiele, erwartet. Durch die Integration von Audio und Video werden Spiele auf dem Copmputer in Zukunft noch realitätsnäher gestaltet werden können. Farbgebung und Layout der Benutzungsoberfläche, Musiklautstärke oder Geschwindigkeit von elektronischen Spielen lassen sich vom Benutzer dynamisch je nach seinen momentanen Wünschen einstellen.
- **Aus- und Weiterbildung, Computer biased training (CBT)**
Zahlreiche Untersuchungen und Erfahrungen haben gezeigt, daß durch multimedial vermitteltes Wissen sowohl eine hohe Lernzeiterparnis als auch ein besseres Verständnis und ein besserer Behaltensgrad erzielt werden kann. Für die Betriebe bedeutet dies eine Reduktion der Schulungskosten. Genauso wie der Unterhaltungsbereich ist auch der Bereich des CBT durch eine hohe Interaktivität der Anwendung gekennzeichnet. Z.B. können je nach Korrektheit der Antworten Lektionen eines elektronischen Sprachkurses wiederholt oder weiterführende Lektionen gestartet werden.
- **Werbung, und Firmenpräsentationen**
Werbung muß effektiv und attraktiv sein, um Verbraucher für bestimmte Produkte zu begeistern. Mit Hilfe von eintönigen Produktpräsentation wird es kaum möglich sein, Produktnamen in die Köpfe der Verbraucher zu bringen. Sogenannte Kiosk-Systeme, auch Point-of-Information/Point-of-Sale-Stationen genannt, ermöglichen nicht nur die Präsentation bestimmter Produkte rund um die Uhr, sondern erlauben sogar nach Geschäftsschluß mittels interaktiver Techniken die Bestellung bestimmter Produkte und sogar deren direkte Bezahlung mittels Kreditkarte. Reisende auf Flughäfen oder Bahnhöfen können sich bei Wartezeiten mit Hilfe multimedialer Auskunftssysteme bereits über ihre Zielgebiete informieren.
- **Kommunikation**
Durch neue Techniken im Bereich der Datenkompression und immer schnellere Netze (Hochgeschwindigkeitsnetze) wird es in die Zukunft möglich sein, elektronisch Post nicht mehr nur textuell sondern mit Sprache und Videosequenzen versehen zu verschicken. Ein anderes Beispiel sind Videokonferenz (Bildtelefon) (effektiver Informationsaustausch).

- elektronische Publikation
- Archivierung und Katalogerstellung
- Dokumentation
- Produktion, Qualitätskontrolle
- Navigationssysteme
- Wartung, Reparatur und Produktservice

So schön und interessant solche Visionen auch klingen, das Problem dabei ist, daß derartige Anwendungen erst einmal entwickelt werden müssen.

5.3 Realisierungsmöglichkeiten

Die Verwendung konventioneller Programmiersprachen stellt natürlich eine Realisierungsmöglichkeit dar. Sie ist jedoch mit einem hohen Zeit- und Kostenaufwand verbunden und nur erfahrenem Programmierern vorbehalten. Entwickler multimedialer Anwendungen sind dahingegen im allgemeinen Designer oder Werbefachleute, die kaum Programmierkenntnisse besitzen. Diesen Personen müssen Software-Werkzeuge zur Verfügung gestellt werden, die eine einfache und schnelle Entwicklung multimedialer Anwendungen ohne Programmierkenntnisse ermöglichen. Derartige Werkzeuge stellen die Autorenwerkzeuge dar. Durch die Benutzung eines Autorenwerkzeugs kann sich der Entwickler voll und ganz der Aufbereitung der zu präsentierenden Informationen und der Gestaltung der Benutzungsschnittstelle der Anwendung widmen und muß sich nicht um die programmiertechnische Realisierung kümmern.

5.4 Definitionen

Ziel einer multimedialen Anwendung ist es, einen Benutzer über einen bestimmten Sachverhalt zu informieren. Dieses Ziel wird dadurch erreicht, daß der Sachverhalt in einzelne voneinander unabhängige Informationseinheiten unterteilt wird, die dann wieder in einen Zusammenhang gebracht werden müssen. Multimediale Anwendungen lassen sich damit als ein Netzwerk aus Knoten und Kanten charakterisieren, ein sogenanntes multimediales Beziehungsnetzwerk. Die Knoten repräsentieren die elementaren Informationseinheiten. Sie werden als Informationsobjekte bezeichnet. Die Kanten legen den Zusammenhang, d.h. Beziehungen zwischen den Informationsobjekten fest. Multimediale Präsentationen bestehen aus einer Menge von Informationseinheiten. Die einzelnen Informationseinheiten kapseln jeweils bestimmte Informationen und können, angelehnt an die Terminologie der objektorientierten Programmierung, als Objekte betrachtet werden. Die Objekte sind von einem bestimmten Typ, genauer gesagt Medientyp, und werden daher als (elementare) Medienobjekte bezeichnet. So sind beispielsweise Texte Instanzen des Medientyps Text und Musikstücke Instanzen des Medientyps Audio.

5.4.1 Informationsobjekte

sind

1. Medienobjekte wie Texte, Graphiken, Ton, Videos, und Animationen.
2. Interaktionsobjekte wie Buttons, Menüs und Texteingabefelder.
3. Anwendungsspezifische Objekte wie eine Datenbank, eine Wissensbasis.
4. Komplexe Informationsobjekte (Multimedia-Objekte): Manchmal es ist sinnvoll ein Teil des Beziehungsnetzwerks als eine Einheit zu betrachten, beispielsweise ein Film mit seiner Vertonung.

5.4.2 Beziehungen

Der Zusammenhang, in dem die einzelnen Informationseinheiten einer multimedialen Anwendung präsentiert werden, läßt sich über Beziehungen zwischen den Informationsobjekten modellieren. Beziehungen können dadurch charakterisiert werden, daß bestimmte auslösende Ereignisse (Auslöser) bestimmte Aktionen (Auswirkung) implizieren. Auslöser und Auswirkung können dabei verschiedenen Typs sein.

Zeitliche Beziehungen:

Über zeitliche Beziehungen wird die zeitliche Abfolge definiert, in der die einzelnen Medienobjekte einem Benutzer präsentiert werden sollen. Beispiele für zeitliche Beziehungen zwischen Medienobjekten sind:

- Ein Video und seine Vertonung sollen gleichzeitig ausgegeben werden.
- Der Start eines Videos oder der Abbruch eines Audios.

Zur Modellierung zeitlicher Beziehungen zwischen Medienobjekten werden drei Ansätze unterschieden:

- Hierarchie Komposition,
- Komposition auf einer Zeitachse,
- Komposition über Referenzpunkte.

Räumliche Beziehungen:

Räumliche Beziehungen legen den geometrischen Zusammenhang, d.h. Position und Größe, zwischen Medienobjekten fest. Sie können daher nur zwischen visuellen Medienobjekten definiert werden. Beispiele für räumliche Beziehungen sind:

- Zwei Bilder sollen in demselben Fenster präsentiert werden.
- Ein Bild soll über ein anderes Bild gelegt werden, falls sie sich überdecken.

In existierenden Autorenwerkzeuge Beziehungen von besonderer Bedeutung sind die sogenannten initialen (diskreten) räumliche Beziehungen. Das sind räumliche Beziehungen, die nur beim Start des Objektes, zu dem die Beziehung hinführt, erfüllt sein müssen.

Gestalterische Beziehung:

Medienobjekte beinhalten eine bestimmte Information, d.h. sie definieren, was einem Benutzer präsentiert wird. Während zeitliche Beziehungen festlegen, wann die Informationen präsentiert werden, werden gestalterische Beziehungen dazu benutzt anzugeben, wie die Information relativ zu anderen Objekten gesehen dargestellt wird.

Gestalterische Beziehungen werden nicht zwischen den Objekten selbst definiert, sondern zwischen Attributen der Objekte, deren aktuelle Werte das Präsentationsbild bestimmen. Beispiele für gestalterische Beziehungen sind:

- Die Ablaufgeschwindigkeit eines Filmes soll immer gleich der Geschwindigkeit seiner Vertonung sein.
- Die Farbgebung eines Graphik-Objektes soll abhängig sein von der Größe des Objektes.

Interaktionsbeziehungen:

Interaktionsbeziehungen sind Beziehungen, die von Interaktionsobjekten ausgehen. Beispiele für Interaktionsbeziehungen sind:

- Das Drücken eines Buttons soll den Start eines Videos bewirken.
- Mit Hilfe eines Schiebereglers soll die Lautstärke eines Audios reguliert werden.

5.5 Entwicklungsprozeß

In diesem Abschnitt wird der Prozeß vorgestellt, der der Entwicklung einer multimedialen Anwendung zugrundeliegt. Der Entwicklungsprozeß kann in mehrere Phasen zerlegt werden. Diese Zusammenhänge werden in (Bild 5.1) graphisch dargestellt und im folgenden analysiert.



Abbildung 5.1: Entwicklungsprozeß [Bol94]

5.5.1 Entwicklungsphasen

Die Entwicklung einer multimedialen Präsentationsanwendung erfolgt in mehreren Phasen, die jedoch im allgemeinen nicht sequentiell durchlaufen werden, sondern denen ein iterativer Entwicklungsprozeß zugrundeliegt (siehe Bild 5.1 [Bol94]).

5.5.1.1 Medienobjekterzeugungsphase

In der Medienobjekterzeugungsphase werden die elementaren Medienobjekte, d.h. Texte, Musikstücke, Videosequenzen, oder auch spezielle Interaktionsobjekte, wie konkrete Menüs, buttons, entweder neu erstellt oder aus bereits existierenden Bibliotheken kopiert. Es wird eine Sammlung von Medienobjekten (Medienobjektsammlung) erstellt, zwischen denen noch keinerlei Beziehungen existieren. Zur Erzeugung der elementaren Informationseinheiten werden informationsspezifische Werkzeuge benötigt (Informationsobjekterzeugungswerkzeuge). Dies geschieht z.B. mit Texteditoren, Graphikeditoren und Animationstools.

5.5.1.2 Verbindungsphase

Die Verbindungsphase setzt sich aus den drei Phasen layout-Phase, Strukturierungsphase und Test-Phase zusammen (siehe Bild 5.2).

Layout-Phase:

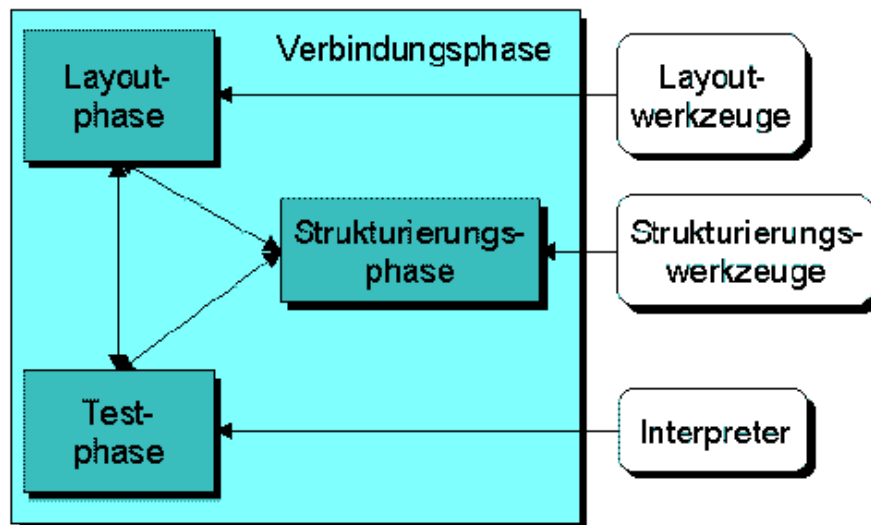


Abbildung 5.2: Verbindungsphase [Bol94]

Medienobjekte repräsentieren eine bestimmte Information. Dies wird aber nicht nur aus dem gebildet, was präsentiert werden soll, vielmehr ist auch die Art, wie der Inhalt dargestellt wird, von Bedeutung. So kann der Informationsgehalt eines Musikstückes, das leise vorgespielt wird, ein anderer sein, als wenn dasselbe Musikstück laut wiedergegeben wird. In der Layout-Phase werden die initialen Eigenschaften der einzelnen Informationsobjekte, wie beispielsweise die Position einer Graphik auf dem Bildschirm, festgelegt. Das geschieht, indem bestimmte Attribute (Gestaltungsattribute) initialisiert werden. Gestaltungsattribute sind häufig zum Beispiel das Fenster, in dem das Objekt dargestellt werden soll, und die Position und Größe des Objektes im Fenster oder auch die Lautstärke eines Audio-Objektes.

Strukturierungsphase

Hier werden Beziehungen zwischen den Objekten definiert. Von besonderer Bedeutung sind dabei die zeitliche Beziehungen, da sie das Ablaufverhalten der Präsentation bestimmen.

Testphase

In der Testphase kann der Entwickler bereits erstellte Teilausschnitte der Gesamtrepräsentation anschauen und damit überprüfen, ob die Präsentation seinen Vorstellungen entspricht. Falls dies nicht der Fall ist, kann er unmittelbar Änderungen am Layout oder der Beziehungsstruktur vornehmen. Das Ergebnis dieser Phase ist der Beziehungsnetzwerk.

5.5.1.3 Generierungsphase

Es wird aufbauend auf dem Beziehungsnetzwerk ein editierbarer Programmcode erzeugt.

5.5.1.4 Programmierphase

In der Verbindungsphase werden häufig graphisch-interaktive Werkzeuge eingesetzt, mit denen sich im allgemeinen nicht alle Eigenschaften einer multimedialen Anwendung spezifizieren lassen. In der Programmierphase kann der in der Generierungsphase erzeugte Programmcode editiert und die multimediale Anwendung in ihre endgültige Form gebracht werden. Der geänderte Code wird anschließend in

ein ausführbares Programm übersetzt.

5.5.2 Entwicklungswerkzeuge

Die Layout-Werkzeuge, Strukturierungswerkzeuge, Interpreter, Codegenerator und die Programmierumgebung werden im allgemeinen zu einer Einheit, den sogenannten Autorenwerkzeugen oder Autorensystem zusammengefaßt. Ziel der Autorensysteme ist es, den Entwicklungsprozeß interaktiver multimedia-ler Anwendungen durch graphisch-interaktive Hilfsmittel (WYSIWYG-Prinzip: What you see is What you get) zu unerstützen und einem Autor den Umgang mit den neuen Möglichkeiten der elektronischen Publikation zu erleichtern.

5.6 Elektronisches Publizieren

Der gesamte Publikationsprozeß läßt sich in drei Phasen unterteilen:

- Konstruktionsphase:**
Informationen werden vom Autor aufbereitet und in Form von Dokumenten zur Verfügung gestellt.
- Distributionsphase:**
Umfaßt den Zeitraum zwischen der Erstellung eines Dokuments bis zu seiner Verfügbarkeit für den Leser. Traditionell sind in dieser Phase Verlage, Herausgeber, Gutachter und weitere Organi-sationen, wie Online-Hosts oder Buchhandlungen, involviert.
- Rezeptionsphase:**
Der Leser nimmt die Informationen auf.

Man spricht von elektronischen Online-Publizieren, wenn der Publikationsprozeß eines Dokumentes so-wohl durch eine elektronische Konstruktion als auch durch eine elektronische Distribution als auch durch eine elektronische Rezeption gekennzeichnet ist. Als elektronishe Konstruktion wird dabei die Dokumentenerstellung mit Hilfe von Computern bezeichnet. Resultat sind in elektronischer Form vor-liegende Dokumente. Unter elektronischer Distribution versthet man die Verbreitung von Dokumenten über Rechnernetze anstelle über nicht-elektronische Distributionkanäle, wie der Post. Elektronische Rezeption bedeutet eine EDV-gestütze Informationsaufnahme durch den Leser gegenüber einer nicht-elektronischen Rezeption beispielsweise via Papier.

Beim elektronischen Offline-Publizieren kann die Distribution der in elektronischen Form liegenden Do-kumente nicht elektronisch erfolgen. Beispielsweise können die elektronischen Dokumente auf Diskette oder CD-ROM mittels der Post verschickt und beim Leser wieder in einem Rechner eingespielt werden.

Konstruktionsphase	elektronisch	elektronisch	elektronisch
Distributionsphase	nicht elektronisch	elektronisch	nicht elektronisch
Rezeptionsphase	elektronisch	elektronisch	nicht elektronisch
	elektr. Offline-Publik. z.B. CD-ROM-Katalog	elektr. Online-Publik. z.B. WWW-Seite	z.B. Bücher

Abbildung 5.3: Elektronische Offline-/Online-Publikation

5.6.1 Autorensysteme für die elektronische Offline-Publikation

Bezüglich der Art und Weise, wie zeitliche Beziehungen zwischen den Informationsobjekten spezifizierbar sind, lassen sich Autorensysteme in drei Klassen einteilen:

1. Framebasierte Autorensysteme:

Die zu repräsentierenden Objekte sind auf Flächen angeordnet. Diese Flächen werden als Frames, Karten, Seiten, Fenster oder auch Dias bezeichnet und sie repräsentieren im Prinzip einen Bildschirm, wie ihn ein Benutzer während der Präsentation für einen bestimmten Zeitraum zu sehen bekommt. Eine komplette interaktive multimediale Anwendung setzt sich aus einer Menge solcher Flächen zusammen, die einen Benutzer in einer bestimmten von ihm durch Interaktion beeinflussbaren Reihenfolge gezeigt werden.

Die Menge der Objekte eines Frames bildet ein komplexes Informationsobjekt. Spezielle Objekte einer Karte können als Anker markiert werden. Navigationsinteraktionen bewirken einen Wechsel der gerade präsentierten Seite.

Bsp. für framebasierte Autorensysteme:
Toolbook für PC, HyperCard für Macintosh.

2. Timeline-basierte Autorensysteme:

In timeline-basierten Autorensystemen werden die Informationsobjekte symbolisch auf einer Zeitachse angeordnet, die den zeitlichen Verlauf der Präsentation festlegt.

Die räumliche Anordnung der Objekte auf dem Bildschirm geschieht wie bei Frame-basierten Autorensystemen direkt-manipulativ bzw. über Menüs. Navigationsinteraktionen können der Zeitachse zugeordnet werden und bewirken im allgemeinen einen Zeitsprung.

Bsp. für timeline-basierten Autorensysteme:
Macromind Director für PC und Macintosh.

3. Flowchart-basierte Autorensysteme:

Hier werden Multimedia Anwendungen durch Flußdiagrammen repräsentiert. Informationsobjekte werden durch Ikonen repräsentiert, in Diagrammen durch Kanten miteinander verbunden, die den möglichen Verlauf widerspiegeln. Gehen dabei von einem Objekt mehrere Kanten zu unterschiedlichen Objekten aus, so wird die während der Präsentation tatsächlich durchflossene Kante im allgemeinen durch eine Navigationsinteraktion bestimmt.

Bsp. für Flowchart-basierte Autorensysteme:
Authorware Professional.

5.6.2 Autorensysteme für die elektronische Online-Publikation

Das World Wide Web ist zur Zeit, die am meisten benutzte Möglichkeit (Methode) zum elektronischen Online-Publizieren. Im WWW werden Informationen auf dem Internet sowie lokale Informationen, die über das Netz zugreifbar gemacht werden sollen, in Form von hypertext/Hypermedia - Dokumenten organisiert.

Navigieren im Netz, geschieht über die Hyperlinks, die von einem Dokument auf ein anderes verweisen. Die Dokumente können dabei physisch auf verschiedenen Rechnern verteilt sein. Zum Auffinden bzw. zum Lesen von Dokumenten dienen WWW-Clients, auch Browser genannt (wie Netscape). Um ein neues Dokument in das WWW einzuspielen, müssen die Dokumente in einem bestimmten Dokumentenformat, der Hypertext Markup Language (HTML) vorliegen. Dazu gibt es eine Vielzahl von HTML-Editoren.

5.7 Hypertext/Hypermedia

5.7.1 Motivation für die Entwicklung

Das menschliche Gedächtnis ist assoziativ. Wir verbinden kleine Informationsstücke mit einander und erzeugen komplexe Wissenstrukturen. Meistens erinnern wir uns an die Informationen auch in assoziativer Weise. Beim Lesen oder beim Speichern von Informationen versucht man die Informationen mit schon vorhandenen Informationen zu verbinden. Die Reihenfolge, wie ein Mensch eine Idee mit einer anderen verbindet hängt von dem Kontext, in dem er gerade arbeitet ab. Durch Hypertext/Hypermedia haben wir die Möglichkeit zur Erzeugung nicht linearer Informationsverkettungen, die die Art und Weise der menschlichen Wissensrepräsentation viel besser nachbilden als traditionelle lineare Informationsverkettung [GLR95]. Sequentielle Strukturen, wie herkömmliche Bücher, spiegeln nicht die Art und Weise wider, wie der Mensch Wissen in seinem Gehirn repräsentiert, nämlich als komplexes Netzwerk. Beim Lesen müssen komplexe Zusammenhänge mit Hilfe von Indexen wiederhergestellt werden. Bei dieser Art von Informationsaustausch geht zwangsläufig Wissen verloren. Hypertext/Hypermedia-Systeme ermöglichen dahingegen den Aufbau netzwerkartiger und damit natürlicher Informationsstrukturen.

5.7.2 Charakteristische Merkmale

Knoten (Hyperknoten) repräsentieren Informationseinheiten. Kanten (Hyperlinks) repräsentieren Verbindungen bzw. Zusammenhänge zwischen den Knoten. Sie sind gerichtet. Hauptsächlich werden sie dazu benutzt, auf ergänzende Informationsquellen zu verweisen.

5.8 Bewertung

Das Publizieren von Informationen mittels elektronische Hilfsmittel bringt Vorteile als auch Nachteile mit sich (Abbildung 5.8).

+	Multimediale Systeme sehen in Ihrem Benutzer keinen passiven Komponenten sondern einen aktiven Teilhaber, der die Anwendung durch Eingaben beeinflussen kann.
+	Besseres vermitteltes Wissen durch die Nutzung von verschiedenen Medien.
-	Bildschirmqualität: Es gibt viele Benutzer die dem Lesen vom Papier das Lesen von Bildschirm vorziehen.
-	Informationsfilterung: Es müssen Werkzeuge entwickelt werden, die dem Benutzer Hilfen die relevante Informationen auszufiltern.
-	Eingabegeräte: Es müssen Eingabegeräte entwickelt werden, die die Multimedia-Anwendungen besser unterstützen, wie Spracheingabe (Spracherkennung) oder Datenhandschuhen.

Abbildung 5.4: Vor- und Nachteile des elektronischen Publizierens.

5.9 Literatur

- [Bol94] Dietrich Boles. Das IMRA-Modell. URL: <http://www-is.informatik.uni-oldenburg.de/~diploma-thesis>, Universitaet Oldenburg, 1994.

-
- [Bol95] Dietrich Boles. Elektronisches Publizieren. In W. Neubauer, Hrsg., *Deutscher Dokumentartag 1995*, Seiten 393–411. Deutsche Gesellschaft für Dokumentation, URL: <http://www-is.informatik.uni-oldenburg.de/~dibo>, Sept. 1995.
- [GLR95] Athula Ginige, David B. Lowe und John Robertson. Hypermedia Authoring. *IEEE Multimedia*, 2(4):24–35, Winter 1995.

Kapitel 6

Layout-/Design-Richtlinien für elektronische Dokumente

Heiko Roth

6.1 Übersicht

Im heutigen Computerzeitalter werden immer mehr elektronische Dokumente erstellt. Vielen Autoren fehlen aber Design-Grundkenntnisse. Die Ergebnisse sind entsprechend schlecht. In den folgenden Kapiteln werden deswegen Layout- und Design-Richtlinien für elektronische Dokumente besprochen. Die Aufteilung der Kapitel wird im folgenden kurz besprochen.

Im Kapitel Grundlegendes wird erklärt, was zum Designbeginn zu beachten ist. Danach werden allgemeine Regeln zur Sprache und Gestaltung beschrieben. Wichtig ist immer: Inhalt und Form müssen zusammenpassen. Die Kapitel Typografie und Layout gehen dann näher auf die Gestaltung von Schrift bzw. die Seitengestaltung ein. Diese beziehen sich auf gedruckte Dokumente, werden aber auch bei elektronischen Dokumenten benötigt. Im nächsten Kapitel werden die Besonderheiten elektronischer Dokumente behandelt. Zum Schluß wird noch kurz das World Wide Web behandelt, das heute die große Bühne für elektronische Dokumente überhaupt ist.

6.2 Grundlegendes

6.2.1 Ausgangssituation

Vor dem eigentlichen Designbeginn ist eine Analyse der Ausgangssituation durchzuführen. Der Designer muß feststellen, welche Information der Veröffentlicher anbieten möchte und welches Medium (Broschüre, Wurfsendung, Internet, ...) verwendet werden soll. Je nach Art des Mediums muß der Designer unterschiedliche Texte und Grafiken erstellen.

6.2.2 Zielgruppe

Vor dem Design ist eine Zielgruppenanalyse notwendig. Diese ist wichtig, um Inhalt und Erscheinungsform an den Leser anzupassen und nicht an diesem vorbei zu produzieren. Ziel eines Dokumentes ist es, möglichst viele Leser einer bestimmten Zielgruppe anzusprechen.

6.2.3 Ziel

Weiterhin ist vor dem Designbeginn zu überlegen, was das Dokument erreichen soll. Je nach Zielsetzung ist das Dokument dann unterschiedlich zu gestalten. Folgende Zielsetzungen sind denkbar, wobei diese Auflistung nicht vollständig ist:

- Verkaufen
- Informieren

- Unterhalten
- Support
- Werben

6.3 Allgemeine Regeln

Für das Design von gedruckten Dokumenten wurden allgemeingültige Regeln gefunden. Diese Regeln basieren auf psychologische Beobachtungen und Tests. Sie sind teils jahrzehnte alt und immer noch aktuell.

Auch im elektronischen Zeitalter haben sie ihre Gültigkeit nicht verloren. Leider werden sie all zu oft nicht beachtet. Die Regeln teilen sich in zwei Kategorien auf: Sprache- und Gestaltgesetze, auf die in den folgenden Abschnitten eingegangen wird.

6.3.1 Sprache

Schon bei der Gestaltung des Textinhaltes sind einige Grundregeln zu beachten. Der Sprachstil muß auf den Inhalt des Textes sowie dessen Zielgruppe und Zielsetzung ausgerichtet sein. Eine Werbebroschüre für ein Produkt hat z.B. einen völlig anderen Sprachstil als das technische Datenblatt des Produktes. Ziel ist es, dem Leser den Inhalt des Textes auf Anhieb zu vermitteln, so daß dieser nicht lange über den Text grübeln muß, um seinen Inhalt zu verstehen. Der Leser darf außerdem nicht überfordert werden. Die Informationen sollen deshalb in kleinen Häppchen dargereicht werden. Dabei sind folgende Probleme zu beachten [Sch94]:

- die Komplexität der natürlichen Sprache
- Assoziationsmöglichkeiten
- Fachausdrücke sind meist für Laien unverständlich
- die Semantik von Begriffen ist stark kontextabhängig

Ein weiteres Problem entsteht, wenn man einen Sachverhalt möglichst korrekt und vollständig beschreiben will. Dies führt leicht zu geschraubten und unverständlichen Formulierungen. Doch was nützt ein wohlgestalteter Text, wenn der Inhalt sprachlich so schlecht dargestellt wird, daß ihn niemand verstehen kann?

Tausch, Langer und Schulz von Thun [LvTT74] haben folgende vier Gesetze gefunden, deren Anwendung zu einer besseren Textverständlichkeit führen soll¹:

Einfachheit

Die sprachliche Formulierung (der Satzbau und die Wortwahl) soll einfach gehalten werden. Kurze und einfache Sätze sind also zu bevorzugen. Außerdem sollen geläufige Wörter verwendet werden. Fachbegriffe wiederum müssen erklärt werden.

Beispiel:

kompliziert: “Raub ist dasjenige Delikt, das jemand durch Entwendung eines ihm nicht gehörenden Gegenstandes unter Anwendung von Gewalt oder von Drohungen gegenüber einer anderen Person begeht, sofern die Intention der rechtswidrigen Aneignung besteht.”

einfach: “Jemand nimmt einem anderen etwas weg. Er will es behalten. Aber es gehört ihm nicht. Beim Wegnehmen wendet er Gewalt an oder droht dem andern, daß er ihm etwas Schlimmes antun werde. Dieses Verbrechen heißt Raub.”

¹Die Beispiele sind aus [LvTT74] übernommen worden.

Gliederung und Ordnung

Der Text soll geordnet und übersichtlich aufgebaut sein, d.h. der Aufbau des Textes soll durch Absätze und Überschriften sichtbar gemacht werden (*äußere Gliederung*), die Sätze sollen folgerichtig aufeinander bezogen sein und die Information in einer sinnvollen Reihenfolge darbieten (*innere Ordnung*), die Unterscheidung vom Wesentlichen gegenüber weniger Wichtigem wird durch Hervorhebungen dargestellt. Das ist vor allem bei längeren Texten wichtig.

Beispiel:

ungeordnet: “Jemand wendet gegen einen anderen Gewalt an. Das ist Raub, es gehört ihm nämlich nicht. Er will es für sich behalten, was er ihm wegnimmt. Zum Beispiel ein Bankräuber, der dem Angestellten mit der Pistole droht. Auch wenn man jemanden bedroht, daß man ihm etwas Schlimmes antun will, ist es Raub.”

geordnet: “Raub ist ein Verbrechen: Jemand nimmt einem anderen etwas weg, was ihm nicht gehört. Er will es behalten. Dabei wendet er Gewalt an oder droht dem anderen etwas Schlimmes an. Drei Dinge sind wichtig:

1. etwas wegnehmen, was einem nicht gehört
2. es behalten wollen
3. Gewalt oder Drohung

Beispiel: Ein Bankräuber droht dem Angestellten mit der Pistole und nimmt sich das Geld.”

Kürze und Prägnanz

Es muß das ideale Verhältnis zwischen Sprachaufwand und Lehrziel gefunden werden. Unnötiger Sprachaufwand entsteht durch nicht notwendige Einzelheiten, Zusatzinformationen und Erläuterungen, breites Ausholen und Abschweifungen vom Thema. Auch sprachliche Entbehrlichkeiten wie weitschweifige Formulierungen, unständliche Erklärungen, Wiederholungen, Füllwörter und Phrasen erzeugen einen erhöhten Sprachaufwand.

Zusätzliche Stimulanz

Der Autor soll einen Text anregend, interessant, abwechslungsreich und persönlich gestalten. Der Text soll beim Leser Interesse, Anteilnahme und Lust am Lesen hervorrufen. Die möglichen Mittel sind: Ausrufe, wörtliche Anrede, rethorische Fragen zum Mitdenken, lebensnahe Beispiele, Reizwörter und witzige Formulierungen. Zusätzliche Stimulanz kann aber eine Gliederung unüberschaubar machen und widerspricht dem Gesetz der Kürze und Prägnanz, weswegen dieses Mittel nur in Maßen und wohlbedacht eingesetzt werden sollte.

Beispiel:

einschläfernd: “Jemand nimmt einem anderen etwas weg. Er will es behalten, obwohl es ihm nicht gehört. Beim Wegnehmen wendet er Gewalt an oder er droht dem anderen, daß er ihm etwas Schlimmes antun werde. Dieses Verbrechen heißt Raub. Raub wird mit Gefängnis oder Zuchthaus bestraft.”

stimulant: “Nimm an, du hast keinen Pfennig in der Tasche. Aber was ist das? Da geht eine alte Dame mit ihrer Handtasche über die Straße. Du überlegst nicht lange: ein kräftiger Schlag auf ihren Arm und schon bist du mit der Tasche auf und davon. ‘Haltet den Dieb!’, ruft die Dame, weil sie es nicht besser weiß. Richtiger müßte es heißen: ‘Haltet den Räuber!’, denn wenn man dabei Gewalt anwendet oder Drohungen ausstößt, dann ist es Raub. Und wie endet die Geschichte? Nun, meistens endet sie im Knast.”

Die Qualität des fertigen Textes kann durch Versuchspersonen der angepeilten Zielgruppe kontrolliert werden, indem die Wirkung des Textes auf die Versuchspersonen mit dem gewünschten Ziel verglichen wird.

6.3.2 Gestaltgesetze

Die Gestaltgesetze machen deutlich, welche Erscheinungen auf welche Weise als Einheit in räumlicher oder zeitlicher Ausdehnung erlebt werden.

Es gibt über 100 Gestaltgesetze, von denen die wichtigsten hier aufgezählt werden [Sch94]:

Prägnanz und gute Gestalt

Die menschliche Wahrnehmung zeigt eine Tendenz zur guten, ausgezeichneten, deutlichen und prägnanten Gestalt. Dies ist mit der Entwicklungspsychologie, dem Aufbau des menschlichen Gehirns und dem Denkprozeß des Menschen zu begründen.

Prägnante Formen sind zum Beispiel der Kreis, das Dreieck oder das Rechteck. Das sind Formen, die uns täglich begegnen und die einfach aufgebaut sind, weswegen sie uns im Gedächtnis bleiben und sie schnell zu erkennen sind.

Nähe

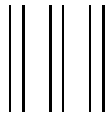


Abbildung 6.1: Nähe

Elemente, die nahe beieinander liegen werden als zusammengehörend wahrgenommen. So werden z.B. in Abbildung 6.1 3 Linienpaare im Gegensatz zu 6 Linien erkannt.

Geschlossenheit

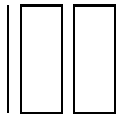


Abbildung 6.2: Geschlossenheit

Geschlossene Elemente werden als zusammengehörend wahrgenommen. In Abbildung 6.2 befinden sich zwei Rechtecke, die jeweils eine Einheit bilden. Außerdem wird deutlich, daß die Gestaltgesetze sich widersprechen können. In dieser Abbildung konkurrieren z.B. Nähe und Geschlossenheit.

Abbildung 6.4: Schriftarten [Sie91, JMH96a]

Jede Schrift läßt sich zunächst einer Schriftart zuordnen (siehe Abbildung 6.4). Hierbei unterscheidet man Antiqua-, gebrochene und Effektschriften. Die Antiquaschriften werden vor allem in langen Texten eingesetzt. Die serifenbehafteten Schriften lassen sich hierbei am besten lesen, da die Serifen die Textzeile verstärken und die Augenführung unterstützen.

Jeder Schrift lassen sich ein oder mehrere Schriftschnitte zuordnen. Darunter versteht man die Variation der Attribute Strichstärke, Schriftbreite und Schriftlage, wie in Abbildung 6.5 zu sehen ist.

Ebenso kann die Schriftlaufweite, das ist der Abstand zweier Buchstaben (siehe Abbildung 6.6) und die Schriftgröße, das ist die Ausdehnung der Buchstaben (siehe Abbildung 6.7) gewählt werden.

Abbildung 6.7: Schriftgröße

Abbildung 6.11: Bildhafte Überschriften

Prospektüberschriften werden meist bildhaft gestaltet. Dann muß die textliche Aussage nicht unbedingt auf Anhieb erkannt werden, denn eine bildhaft gestaltete Überschrift sollte sich in erster Linie durch Originalität auszeichnen (siehe Abbildung 6.11). Anwendung von bildhaften Überschriften ist die Identifikation von Serien in Zeitungen und Zeitschriften.

6.4.4 Satzarten

Ein weiteres wichtiges Layoutkriterium bei Textpassagen ist die Schriftausrichtung oder Satzart. Es gibt vier verschiedene Typen:

- linksbündig
- zentriert
- rechtsbündig
- Blocksatz

Für lange Textpassagen kommen hierbei nur linksbündig und Blocksatz in Frage, was mit unseren Lesegewohnheiten zusammenhängt (von links nach rechts lesen). Blocksatz und linksbündiger Satz sind am kompaktesten und werden deshalb bei langen Texten eingesetzt. Meist wird der Blocksatz bevorzugt, da dieser am einfachsten zu formatieren ist.

Rechtsbündiger und linksbündiger Satz bezeichnet man als Flattersatz; zentrierter Satz wird auch Mittelachsensatz genannt. Beim rechtsbündigen und Mittelachsensatz läßt sich der Zeilenanfang der nächsten Zeile nur schwer finden. Außerdem ist die Formatierung von Flattersatz und Mittelachsensatz aufwendig, wenn Dynamik in den Text eingebracht werden soll. Wenn man hochwertige Dokumente entwerfen möchte, ist hierauf ein besonderes Augenmerk zu richten, denn der Leser empfindet beispielsweise Texte deren Layout eine treppenförmigen Randverlauf aufweisen als ermüdend und uninteressant.

Der Flattersatz, im besonderen der rechtsbündige und zentrierte Satz, sind also nur für Dokumente mit wenig Text oder für Hervorhebungen interessant (z.B. für Überschriften).

6.5 Layout

Layout bezeichnet die Seitengestaltung. Hierzu gehört das Positionieren der verschiedenen Elemente wie Extrahemen, Grafiken und Bildern, sowie die farbliche Gestaltung des Dokuments. Die Positionierung der einzelnen Elemente soll hierbei den Gestaltungsraum zu einer Einheit verschmelzen lassen.

Das Seitenlayout soll im gesamten Dokument einheitlich sein, so daß die Einheit des Dokuments betont und Zusammengehörigkeit der einzelnen Abschnitte und Kapitel erkannt wird. Wichtig ist wieder, wie bei der Typografie, daß sich Inhalt und Form entsprechen.

Alle Elemente sind zu gewichten (Größe und Position), um zu einem harmonischen Ganzen zu kommen. Am besten gelingt das, indem man eine Seite in Rahmen unterteilt. Darunter versteht man ein Raster, an dem im Layoutprozeß Textelemente und Bilder ausgerichtet werden. Nach diesem Prinzip gestaltete Dokumente wirken in ihrem Erscheinungsbild deshalb besonders homogen.

Weitere Strukturmittel z.B. sind Überschriften, Absätze, Hervorhebungen oder farbige Unterlegungen. Sie sollen dem Leser die Gliederung des Dokuments verdeutlichen.

Überschriften und Absätze

teilen dem Leser kurz mit, welches Thema ihn erwartet.

Hervorhebungen

sind topografische Auszeichnungen durch größere Schrift, anderen Schrifttyp, Fett- oder Kursivdruck.

Unterlegungen

mit Farbe, Mustern oder Grafiken können Textteile hervorheben (Achtung: Auf die Lesbarkeit ist hierbei besonders zu achten!).

Strukturierte Aufzählungen

verdeutlichen optisch Einzelaussagen, ein zusätzliches Aufzählungszeichen erhöht den Effekt.

Signalwörter und Symbole

können dem Leser eine wesentliche Orientierungshilfe sein (z.B. Achtung, Hinweis).

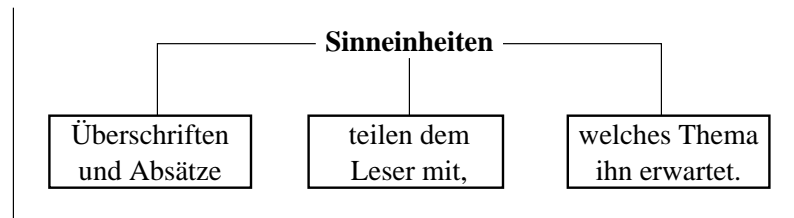


Abbildung 6.12: Das Bilden und optische Trennen von Sinneinheiten sorgt für Aufmerksamkeit und Übersicht

Bei kurzen Texten, wie Anmerkungen oder Hinweisen, können Sinneinheiten gebildet und optisch deutlich getrennt werden, wie das beispielsweise in Abbildung 6.12 zu sehen ist.

Eine Seite wird durch Grafik aufgelockert. Jede Grafik hat ihre eigene Anmutung, wobei ein Foto im allgemeinen mehr Stimmung erzeugen kann als eine Strichgrafik. Gleichzeitig ist eine Strichgrafik oft aussagekräftiger als ein Foto. Wichtig bei Grafiken ist, Kontraste zu wählen, die im Idealfall auch in einer Schwarz/Weiß-Reproduktion noch erkennbar und lesbar bleiben.

Auf folgendes ist im besonderen zu achten:

- Ausgabe und Präsentationsgeräte
- Lesegewohnheiten
- Augenführung, welche dem Ablauf des Dokuments entspricht
- nicht überfrachten, sondern wenig Information (möglichst nur eine) in den Vordergrund stellen (beschränktes Kurzzeitgedächtnis des Menschen)

Ein weiteres Gestaltungsmittel im Layout ist die Farbe. Dabei sollte man berücksichtigen, daß jede Farbe ihre eigene psychologische Wirkung (z.B. rot: Gefahr) hat. Wie auch bei der Schriftwahl sollte man sich auf wenige Farben beschränken und diese durchgängig und sinnvoll einsetzen, um auf Wichtiges hinzuweisen. Hier gilt wieder, daß zuviel den gewünschten Effekt abschwächt und die Übersicht stört. Zum Abschluß dieses Kapitels noch ein Tip, wie man das richtige Layout für ein Dokument am besten findet. Da ein Dokument phasenweise entsteht, sollte man jede Gestaltungsphase ausdrucken und die Einzelergebnisse nebeneinander legen. Der Vergleich der Ausdrucke schult das Gefühl für die Form und mögliche Verbesserungen fallen einem besser ins Auge.

6.5.1 Corporate Design

Heutzutage werden von einem Designer Gestaltungsrichtlinien innerhalb einer Firma festgelegt. Diese bestimmen das ganze Erscheinungsbild der Firma. Die Lackierung der Firmenwagen, die Außenfassade des Firmengebäudes, die Werbeplakate und Werbespots der Firma usw. sind davon betroffen.

Natürlich erhalten auch interne und externe Firmendokumente ein einheitliches Layout. Diese grundlegende Festlegung des Dokumentlayouts hat viele Vorteile:

gutes Design

Die Vorgaben werden in der Regel von qualifizierten Grafikagenturen erstellt und gewährleisten daher, daß wichtige Details wie gute Lesbarkeit, sinnvolle Verwendung von Farbe und Formen usw. bereits überprüft wurden.

sofort nutzbar

Es muß kein neues Layout entworfen werden.

Wiedererkennung

Ein Dokument kann sofort der Firma zugeordnet werden.

6.6 Elektronische Dokumente

Die bisherigen Kapitel beschäftigten sich mit gedruckten Dokumenten. Die in diesen Kapiteln vorgestellten Regeln gelten aber auch für elektronische Dokumente. Sie können als Weiterentwicklung von Standarddokumenten angesehen werden. Allerdings sollte man schon beim Erstellen eines elektronischen Dokuments auf dessen Besonderheiten wie die Aufteilung in informationelle Einheiten und Verknüpfungen achten.

Für elektronische Dokumente² sind folgende vier Konzepte wichtig, die bei gedruckten Dokumenten in dieser Form nicht anzutreffen sind. Die Konzepte werden in den nächsten Abschnitten erklärt. Sie lauten:

- Informationelle Einheit / Seite
- Nicht-Linearität / Verknüpfungen
- Orientierungs- und Navigationshilfen
- Skripten

6.6.1 Informationelle Einheit (Seite)

Da elektronische Dokumente vor allen Dingen gezielt Information vermitteln sollen, müssen sie in kleine, kompakte Einheiten unterteilt werden, die informationelle Einheiten oder auch Seiten³ genannt werden. Komplexe Information kann so strukturiert dargestellt werden und der Benutzer eines elektronischen

Abbildung 6.14: Nicht-Linearität

Schon Standarddokumente besitzen ein gewisses Maß an nicht-linearen Strukturen. Hierzu zählen: Verweise auf andere Seiten, Abschnitte, Kapitel oder Bücher und die Inhaltsangabe, der Index, Glossar oder ein Lesezeichen.

In elektronischen Dokumenten ist der Anteil nicht linearer Strukturen ungleich höher (siehe Abbildung 6.14), da hier Information vermittelt werden soll, auf die der Benutzer gezielt zugreifen kann.

²Dieses Kapitel ist an Kapitel 2 aus [Rot96] angelehnt.

³Der Begriff Seite darf hier aber nicht mit einer Buchseite verwechselt werden.

Der Zusammenhang (die Kohärenz) eines elektronischen Dokumentes muß durch Verweise (Verknüpfungen) zwischen den einzelnen Seiten entstehen, da der lineare Zusammenhang fehlt. Es besteht die Gefahr, sich in den Verknüpfungen zu verlieren.

6.6.3 Orientierungs- und Navigationshilfen

Von Standarddokumenten sind folgende Orientierungshilfen bekannt:

Inhaltsangabe

Gibt einen Überblick über das Dokument.

Index

Auflistung von Begriffen mit Seitenverweis.

Glossar

Erklärung wichtiger Begriffe bzw. Erklärung von Fachbegriffen.

Lesenzeichen

Dient zum Wiederfinden einer Textstelle.

Diese Orientierungshilfen sind auch in elektronischen Dokumenten anzufinden. Durch den Computer sind sie einfacher zu bedienen als bei gedruckten Dokumenten und wegen der nicht-linearen Struktur elektronischer Dokumente, sind sie für elektronische Dokumente noch wichtiger als für gedruckte Dokumente.

Darüberhinaus haben sich für elektronische Dokumente weitere Orientierungs- und Navigationshilfen entwickelt:

Historie

Auflistung der zuletzt betrachteten Seiten.

Markierung gelesener Seiten

Seiten, die schon betrachtet wurden, werden markiert.

lokales Glossar

Eine Liste von Glossareinträgen speziell für eine Seite. Diese Einträge sind Teil eines globalen Glossars.

Guided Tours (Geführte Unterweisungen)

Ausgewählte Seiten des Dokuments zu einem bestimmten Thema. Diese Seiten sollten einen gewissen Zusammenhang bilden. In einem elektronischen Dokument sind in der Regel mehr als eine Guided Tour möglich.

Übersichtskarte

Ein Graph, der die gesamte Seite und Verweise des Dokuments aufzeigt.

Fischauge

Zeigt die Umgebung einer Seite, d.h. alle Seiten, die auf die betreffende Seite verweisen und alle Seiten, auf die die betreffende Seite verweist.

6.6.4 Skripten

Da elektronische Dokumente auf Computern betrachtet werden, können mit einigen Autorensystemen ganze Programmteile in Dokumente eingefügt werden. Dokumente können dadurch sehr dynamisch und interaktiv gestaltet werden. Der Übergang zu Anwendungen oder Spielen ist fließend.

6.7 World Wide Web

Das Internet ist heutzutage in aller Munde. Meist ist das World Wide Web, kurz WWW damit gemeint. Im WWW tummeln sich Millionen von elektronischen Dokumenten. In folgenden werden kurz die Möglichkeiten und Probleme von Dokumenten beschrieben, die im WWW veröffentlicht werden.

Durch die Popularität des Internet ergeben sich zur Zeit Übertragungsengpässe. Die begrenzte Bandbreite bei der Übertragung ist schon während des Designs eines Dokuments zu berücksichtigen. Die Dokumente dürfen nicht mit Grafiken überfrachtet sein, da sie sonst, je nach der Geschwindigkeit des Zugangs oder Entfernung von der Dokumentquelle, Ewigkeiten zum Laden brauchen und nicht sofort dargestellt werden. Die Dokumente sollten deswegen immer so aufgebaut sein, daß ihr Inhalt im Notfall auch ohne Grafiken erfaßt werden kann.

Weitere Probleme stellen die verschiedenen Standards und Browser dar. Das Erscheinungsbild und die Darstellung der einzelnen Browser ist unterschiedlich. Des weiteren kann ein Browser meist auch noch vom Benutzer frei konfiguriert werden. So können z.B. die Hintergrundfarbe oder die Grundschrift frei gewählt werden.

Außerdem herrscht im Netz eine hohe Dynamik. Damit das Dokument viel gelesen wird, muß der Designer sein Dokument inhaltlich und gestalterisch auf dem laufenden halten, denn auch die Standards, d.h. die Beschreibungssprache HTML, entwickelt sich ständig weiter. Die Dokumente selbst können auch immer dynamischer gestaltet werden. So können Filme und animierte Bilder sowie Programme (Java) in ein Dokument eingebunden werden.

Ein weiterer Vorteil ist der mögliche Feedback, der Online geschieht. Kritik oder Bestellungen können so in Reaktion auf das Dokument vom Leser abgegeben werden.

Spätestens dies kann mit Standarddokumenten nicht erreicht werden.

6.8 Literatur

- [BDKK96] Jan Borchers, Oliver Deussen, Clemens Knrzer und Arnold Klingert. Layout Rules for Graphical Web Documents. *Computers and Graphics*, 20(3), 1996.
- [JMH96a] Thomas Jaspersen, Michael Mielenz und Thomas Himstedt. Gestaltungsgrundlagen der Präsentation. *Online Erfolgreiches Informationsmanagement*, 5:72–74, 1996.
- [JMH96b] Thomas Jaspersen, Michael Mielenz und Thomas Himstedt. Präsentationen im Berufsalltag. *Online Erfolgreiches Informationsmanagement*, 4:79–80, 1996.
- [Kuh91] Rainer Kuhlen. Hypertext: Ein nichtlineares Medium zwischen Buch und Wissensbank. 1991.
- [LvTT74] Inghard Langer, Friedemann Schulz von Thun und Reinhard Tausch. *Verständlichkeit in Schule, Verwaltung, Politik und Wissenschaft*. Ernst Reinhardt Verlag, 1974.
- [Rot96] Heiko Roth. Programmierung eines Autorensystems zur Umsetzung von Regelwerken zur Kläranlagentechnik in Hypertexte. Studienarbeit: Institut für Siedlungswasserwirtschaft, Universität Karlsruhe (TH), Januar 1996.
- [Sch94] Prof. Dr. Alfred Schmitt. Mensch-Maschine-Dialog I. Institut für Betriebs- und Dialogsysteme, Fakultät für Informatik, Universität Karlsruhe, Oktober 1994.
- [Sch96] Norbert Schaffer. Dies sind Gedanken zur WEB-Seitengestaltung mit besonderem Augenmerk auf die Beziehung Leser-Autor. URL: http://www.ilkk.de/scha/scha_96ged.html, November 1996.
- [Sie91] Manfred Siemoneit. *Typografisches Gestalten, Regeln und Tips für die richtige Gestaltung von Drucksachen*. Polygraph, 1991.
- [Wil96] Wilhelm Wilming. Form und Inhalt. *Organisation und Management*, 102:56–57, 1996.



Kapitel 7

Multimedia ToolBook

Eduard Niess

7.1 Übersicht

Mit der zunehmenden Verbreitung elektronischer Medien, vor allem durch die Möglichkeit der Präsentation der Information auf CD-ROMs, wurde der Wunsch nach Software, die dem Autor die Möglichkeit gibt, ohne große Programmierkenntnisse ansprechend wirkende multimediale Anwendungen zu erstellen, immer häufiger geäußert. Dies führte zur Entstehung einer neuen Klasse von Softwarewerkzeugen, die speziell auf dieses Aufgabengebiet zugeschnitten sind, den sogenannten Autorensystemen. Diese Systeme erlauben dem Benutzer auf eine einfache Weise verschiedene Medien in einer multimedialen Applikation zu kombinieren und miteinander zu verbinden: Texte, Musik, Bilder, Animationen usw.

Inzwischen gibt es mehrere große Systeme auf dem Markt, die mit unterschiedlichen Konzepten versuchen, das elektronische Publizieren zu vereinfachen. Im Folgenden wird einer der weitverbreitesten Vertreter dieser Softwarewerkzeuge, das **Multimedia ToolBook 4.0** von **Asymetrix**, besprochen.

7.2 Klasseneinteilung der Autorensysteme

Die einzelnen Autorensysteme verfolgen unterschiedliche Strategien und Konzepte, die bei der Erstellung einer Applikationen verwendet werden können. Die heutigen Systeme werden in drei große Klassen eingeteilt werden. Aber man muß berücksichtigen, daß die Grenzen zwischen diesen Klassen fließend sind.

7.2.1 Frame-basierte Autorensysteme

Bei den frame-basierten Autorensystemen werden die Objekte auf Flächen angeordnet, die einen Bildschirm repräsentieren, wie ihn der Benutzer schließlich zu sehen bekommt. Eine komplette Applikation besteht aus einer Menge solcher Flächen, die in einer bestimmten Reihenfolge durchgesehen werden kann.

Das **Asymetrix Multimedia ToolBook** gehört in diese Klasse der Autorensysteme.

7.2.2 Timeline-basierte Autorensysteme

Bei diesen Systemen wird die Applikation durch eine Zeitachse repräsentiert, auf der die einzelnen Objekte durch den Autor angeordnet werden. Der Leser geht entlang dieser Zeitachse und bekommt nach und nach die Objekte zu sehen.

7.2.3 Flowchart-basierte Autorensysteme

Bei diesen Autorensystemen werden die Objekte durch Ikonen dargestellt, die miteinander durch Pfade verbunden sind. Diese Pfade repräsentieren die möglichen Wege, die der Benutzer in der fertigen Applikation durchlaufen kann.

7.3 Grundlegende Konzepte von Multimedia ToolBook

7.3.1 frame-basiert

Wie bereits gesagt wurde, gehört das System in die Klasse der frame-basierten Autorensysteme. Die Erstellung der einzelnen Frames erfolgt streng nach dem WYSIWYG-Prinzip. Mit wenigen Mausklicks werden Objekte aus einer Tool-Leiste ausgewählt und auf dem Frame positioniert.

Eine mit **Multimedia ToolBook** erstellte Applikation wird Buch genannt. Die einzelnen Frames werden als Seiten bezeichnet. Die Seiten bestehen aus zwei Teilen: aus einem Hinter- und einem Vordergrund. Der Hintergrund kann von mehreren Seiten gleichzeitig benutzt werden. Hier werden die Objekte positioniert, die auf jeder dieser Seiten zu sehen sein sollen, z.B. Navigationsknöpfe und Graphiken. Da ein Buch beliebig viele Hintergründe enthalten kann, die wiederum vom mehreren Seiten gleichzeitig benutzt werden können, ist eine bequeme Möglichkeit zur Kapiteleinteilung gegeben, indem man für jedes Kapitel einen eigenen Hintergrund erstellt. Jedes Buch enthält immer mindestens einen Hintergrund. Der Vordergrund enthält die eigentlichen Informationsobjekte der Seite. Diese Objekte sind von Seite zu Seite unterschiedlich. Jede Seite hat einen eigenen Vordergrund.

Um die Seiten auf dem Bildschirm sichtbar zu machen, gibt es in ToolBook die sogenannten Viewer, die vereinfacht als Fenster der graphischen Oberfläche realisiert sind, und den Inhalt einer Seite darstellen. Jedes Buch hat mindestens einen Viewer, kann aber beliebig viele weitere Viewer enthalten. Eine Anwendung für einen zweiten Viewer ist z.B. das Anzeigen von Hilfsinformationen zu einem Objekt in einem eigenen Fenster.

7.3.2 ereignisorientiert

Das System arbeitet ereignisorientiert, d.h. jede Interaktion mit dem Benutzer wird in der Form eines Ereignisses dargestellt. Es werden dabei folgende Ereignisarten unterschieden

- *Tastaturereignis*: Das Drücken bzw. das Loslassen einer Taste
- *Mausereignis*: Betreten bzw. Verlassen eines bestimmten Bereichs oder Mausklicks
- *Systemereignisse*: Z.B. Starten einer Applikation, Betreten einer Seite usw.
- *benutzerdefinierte Ereignisse*: Vom Autor der Applikation definiert

Die Ereignisorientiertheit geht so weit, daß selbst wenn der Benutzer nichts tut, ein Ereignis erzeugt wird, das *idle*-Ereignis, auf welches entsprechend reagiert werden kann.

7.3.3 Objekthierarchie

Alle Informationsobjekte einer Seite werden in Schichten eingeteilt, wobei sich in jeder Schicht nur ein einziges Objekt befindet. Diese Einteilung ist vor allem bei der Überlappung mehrerer Objekte wichtig:

- Das System kann eindeutig entscheiden, welches Objekt sichtbar ist.
- Das System leitet ein Ereignis an das Objekt weiter, welches sich auf der höchsten Ebene in dieser Hierarchie befindet.

Bei dem Eintreffen eines Ereignisses hat ein Objekt die Wahl zwischen den folgenden drei Möglichkeiten:

- Es kann auf das Ereignis entsprechend reagieren und danach auf das nächste warten.
- Es kann das Ereignis bearbeiten und danach an die nächsttiefere Schicht weitergeben.
- Schließlich kann es das Ereignis einfach ignorieren, wodurch es automatisch zur tieferen Schicht wandert.

Autormodus	Lesermodus
Applikationen erstellen	
Objekte erstellen	
Objekteigenschaften verändern	
Skripte schreiben	
Texte verändern	
edierbare Texte verändern	edierbare Texte verändern
Drucken	Drucken
	Applikationen ausführen

Tabelle 7.1: Autor- / Lesermodus

7.3.4 skript-orientiert

Multimedia ToolBook ist ein sehr stark skript-orientiertes System. Man kann durch die integrierte Skript-Sprache **OpenScript** komplexe Algorithmen implementieren, sogar komplette **ToolBook**-Applikationen erstellen. Man kann zwar bei kleineren Projekten auf **OpenScript** verzichten, aber schon bei etwas größeren Applikationen kommt man um das Erlernen und Anwenden dieser Programmiersprache nicht herum. In einem späteren Kapitel komme ich auf **OpenScript** noch genauer zu sprechen.

7.3.5 Autor-/Lesermodus

Das System unterscheidet zwischen einem Autor, der eine Applikation erstellt, und einem Leser, der das fertige Buch betrachtet. Der Wechsel zwischen diesen beiden Modi ist jederzeit möglich. Dadurch hat der Autor die Möglichkeit, das Ergebnis seiner Arbeit sofort aus der Sicht des Benutzers zu betrachten. Der Autor hat, wie später noch genauer erklärt wird, die Möglichkeit den Wechsel in den Autormodus dem endgültigen Benutzer zu verwehren und dadurch seine Arbeit zu schätzen.

Die Tabelle 7.1 verdeutlicht die Unterschiede in den Möglichkeiten eines Autors und eines Lesers.

7.4 verfügbare Objekte

7.4.1 graphische Objekte

Die einfachsten Objekte sind Linien, Kreise, Rechtecke usw. Diese dienen vor allem zur räumlichen Trennung anderer Objekte. Außerdem können diese Objekte Bereiche definieren, die auf das Betreten bzw. Verlassen der Maus reagieren.

7.4.2 Buttons

Die wichtigsten Interaktionsobjekte in **Multimedia ToolBook** sind die *Buttons*. Diese dienen zur Interaktion mit dem Benutzer, der die Buttons durch Anklicken aktivieren kann. Man unterscheidet drei Arten von Buttons:

- *Pushbuttons* werden vor allem zur Navigation durch die Applikation oder zur Aktivierung bestimmter Vorgänge benutzt.
- *Check-Buttons* werden zum Auswählen einer oder mehrerer bestimmter Optionen aus einer vorgegebenen Menge benutzt.
- *Radio buttons* werden zur Auswahl einer einzigen unter mehreren Möglichkeiten eingesetzt (multiple choice).

Die Buttons können einen Text und ein Bild (Ikone) besitzen, die entweder gleichzeitig oder einzeln benutzt werden können. Für die Ikonen steht ein eigener Editor zur Verfügung.

7.4.3 Texte

Diese Objekte werden zur Anzeige bzw. Abfrage bestimmter schriftlicher Informationen benutzt. Texte sind Bestandteile vieler anderer Objekte, z.B. Beschriftung der Buttons, sodaß das Aussehen dieser Objekte unmittelbar das Aussehen anderer beeinflusst. Man kann mehrere Attribute einstellen: Größe, Farbe, Schriftart und -stil usw.

7.4.4 Hotwords

Dies sind Bestandteile eines Textes, die besonders hervorgehoben sind und beim Anklicken mit der Maus bestimmte Vorgänge auslösen. Eine typischer Vorgang, der durch das Aktivieren eines Hotwords ausgelöst werden kann, ist das Springen zu einer anderen Seite in der Applikation. Durch die Hotwords kann sehr einfach eine kontextsensitive Navigation durch die Applikation realisiert werden.

7.4.5 Listen und Comboboxen

Die Listen und Comboboxen werden zur Auswahl einer Möglichkeit unter mehreren benutzt (multiple choice). Der Unterschied zwischen diesen beiden Objekten besteht darin, daß man bei der Liste die unterschiedlichen Einträge sofort sieht. Im Gegensatz dazu sieht man bei einer Combobox zunächst nur den zuletzt ausgewählten Eintrag. Erst beim Anklicken der Box bekommt man alle zur Verfügung stehenden Möglichkeiten zu sehen.

7.4.6 Menüs

Die *Menüs* werden benutzt, um dem Leser jederzeit die Möglichkeit zu geben, bestimmte Einstellungen vorzunehmen oder Vorgänge auslösen, neue Daten zu laden bzw. zu speichern, Drucken usw. Das Erzeugen von Menüs erfolgt in einem eigenen Editor. In **Multimedia ToolBook** gibt es drei Arten von Menüs:

- *Drop-down* Menüs sind die am häufigsten benutzte Menüart. Diese Menüs stehen am oberen Fensterrand und sind ständig erreichbar.
- *Submenüs* sind Einträge eines drop-down Menüs, bei denen bei der Auswahl eine weitere Liste mit Einträgen erscheint. Diese Liste kann wieder Submenüs enthalten. Durch den Einsatz von Submenüs werden die drop-down Menüs einfacher und übersichtlicher. Allerdings kann bei einer zu großen Schachtelungstiefe die Übersicht verloren gehen.
- *Popup*-Menüs sind kontextsensitive Listen, die nicht ständig sichtbar sind. Diese Menüs erscheinen normalerweise bei dem Klick mit der rechten Maustaste. In Abhängigkeit von dem Ort des Mausklicks, können verschiedene Menüs erscheinen, welche die zu dem Kontext passende Auswahl bieten.

7.4.7 Bilder

Die graphische Informationen werden in *Bildern* dargestellt. Dabei werden vom Autorensystem die meisten gängigen Bildformate unterstützt. Die Verwaltung der Bilder wird von **Multimedia ToolBook** komplett übernommen, d.h. man bindet ein Bild ein und braucht sich keine Gedanken darüber zu machen, wo und wie es abgespeichert wird. Die Originalbilder können nach dem Einbinden wieder gelöscht werden, da sie in der Applikation gespeichert werden.

7.4.8 Gruppen

Mehrere Objekte einer **ToolBook**-Applikation können zu einer Gruppe zusammengefaßt werden. Danach werden diese Objekte wie ein einziges Objekt behandelt, z.B. werden sie alle gleichzeitig verschoben. Da eine Gruppe auf einer eigenen Ebene in der Objekthierarchie steht, können nun alle zusammengefaßten Objekte gleichzeitig auf ein Ereignis reagieren.

7.4.9 Beispiel

Bild 7.1 zeigt einige der in **Multimedia ToolBook** zur Verfügung stehenden Objekte.

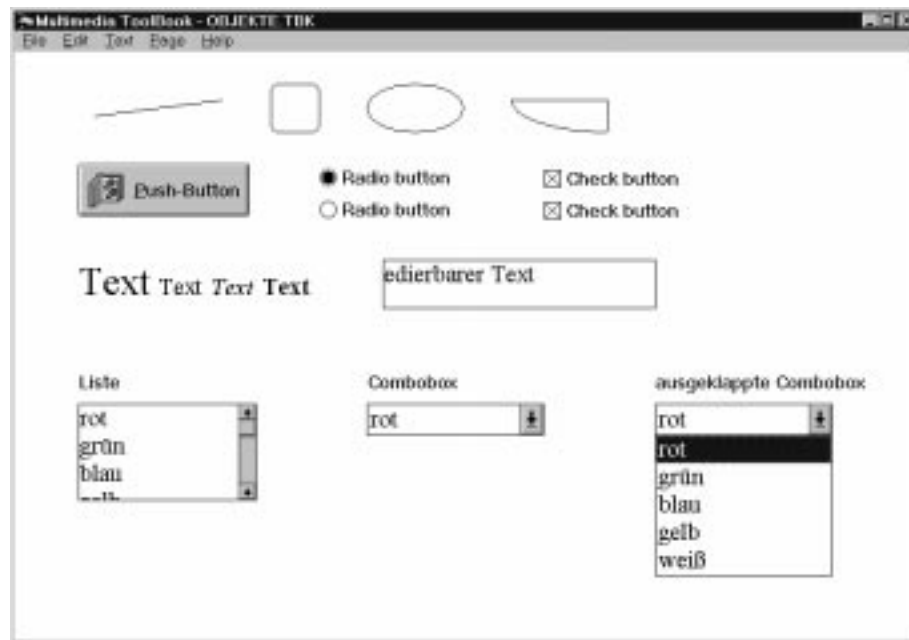


Abbildung 7.1: Beispiele der **ToolBook**-Objekte

7.5 OpenScript

Wie bereits gesagt wurde, ist **Asymetrix Multimedia ToolBook** sehr stark skript-orientiert. Das Erlernen von **OpenScript** ist unverzichtbar, wenn man größere Applikationen erstellen will. Die Sprache ist jedoch sehr leicht erlernbar, vor allem wenn man bereits eine andere Programmiersprache beherrscht. **OpenScript** ist laut Hersteller eine objektorientierte Sprache. Man muß aber berücksichtigen, daß die Sprache nicht alle objektorientierten Paradigmen befolgt. So ist das Vererben oder Definieren neuer Objektklassen nicht möglich. Man hat nur die Objekte zur Verfügung, die der Hersteller definiert hat. Auch die Nichtprogrammierer können die Vorteile einer Skript-Sprache nutzen, ohne sie Erlernen zu müssen, indem sie den *Skript-Recorder* oder *Auto-Skripts* benutzen.

Um den Skript-Recorder zu benutzen, klickt man einfach auf das Aufnahme-Symbol, führt mit der Maus die gewünschten Operationen durch, stoppt die Aufnahme und übernimmt das von **ToolBook** automatisch erstellte Skript. Dieses Skript kann natürlich auch bearbeitet und erweitert werden.

Die Auto-Skripts sind vom Hersteller vorgefertigte Skripte, die viele Fälle der Programmierung abdecken. So existiert z.B. ein Auto-Skript, der einen beliebigen Web-Browser mit einer angegebenen URL aufruft. Es gibt Auto-Skripts zur Navigation, zum Abspielen von WAV-Dateien und viele andere mehr. Diese Skripts können in einem Dialog ausgewählt und einem bestimmten Ereignis zugeordnet werden.

Eine weitere Möglichkeit, Programmierarbeit einzusparen, besteht darin, daß mehrere Objekte ein Skript gemeinsam benutzen können. Man spricht dann von den *shared Scripts*.

Um die Fehler, die sich bei der Programmierung zwangsläufig einschleichen, zu finden und zu korrigieren, existiert in **OpenScript** ein *Debugger*. Dieser Debugger bietet die Möglichkeit die Skripte Schritt für Schritt zu durchlaufen, Variableninhalte zu überwachen und Teile eines Skripts von der Ausführung auszuschließen.

OpenScript verfügt über alle Merkmale einer modernen Programmiersprache. Es folgt eine kurze Übersicht der verfügbaren Elemente des Sprache.

7.5.1 Kommentare

Die Kommentare werden in **OpenScript** durch `--` eingeleitet und gelten bis zum Ende der Zeile, in der sie stehen.

7.5.2 ToolBook-Objekte

Jedes **ToolBook**-Objekt ist in **OpenScript** durch eine eigene Objektklasse vertreten. Man kann jedem Objekt einen Namen geben, um die Handhabung zu erleichtern.

7.5.3 Ereignisse und Handler

Jedes **ToolBook**-Ereignis wird in **OpenScript** durch einen eindeutigen Namen repräsentiert. Die Ereignisse werden durch spezielle Strukturen, die sogenannten *Handler* empfangen. Jeder Handler reagiert auf genau ein bestimmtes Ereignis. Jedes Objekt kann mehrere Handler haben und so auf mehrere verschiedene Ereignisse reagieren.

Durch den Befehl *forward* kann das Ereignis an die nächsttiefere Ebene der Objekthierarchie weitergegeben werden.

Die Grundstruktur eines Handlers sieht wie folgt aus:

```
to handle Ereignis
...
forward -- das Ereignis weiterleiten
end
```

Das Versenden von Ereignissen erfolgt durch das Kommando *send*:

```
send Ereignis [to ...]
```

Nach der Angabe des Ereignisses kann man im *to*-Teil des Befehls das Objekt, welches das Ereignis empfangen soll, und evtl. den Viewer spezifizieren.

7.5.4 Funktionen

Funktionen sind Teile eines Skripts, die verwendet werden, um eine Operation durchzuführen oder ein Ergebnis zu berechnen. **ToolBook** hat über 50 eingebaute Funktionen zur Arithmetik, Statistik, String-Manipulationen und anderen Zwecken. Außerdem kann der Autor eigene Funktionen programmieren. Beim Aufruf einer Funktion kann man dieser Parameter übergeben. Die folgenden Beispiele demonstrieren den Umgang mit den Funktionen:

```
-- ToolBook-Funktion mit Parametern aufrufen und Ergebnis speichern
kState = keyState(keyShift)
-- Definition neuer Funktion
to get neueFunktion param1 param2
return param1 - param2
end
```


7.5.5 Variablen

OpenScript unterstützt *globale* und *lokale* Variablen, die *typisiert* sein können, zur Speicherung von Werten. Die globalen Variablen müssen in jedem Handler und jeder Funktion, die sie verwenden, neu deklariert werden. Lokale Variablen werden nicht deklariert. Sie werden bei der ersten Benutzung erzeugt und beim Verlassen der Funktion oder des Handlers automatisch gelöscht. Dieser Mechanismus kann zu schwer zu findenden Fehlern führen: Wenn der Programmierer vergißt, eine globale Variable zu deklarieren, wird sie automatisch als eine lokale behandelt. Eine Veränderung solcher Variable wirkt sich nicht global aus. Wird dies zu spät bemerkt, ist man gezwungen, die Skripte mehrerer Objekte durchzusehen, was sehr aufwändig sein kann. Die Tippfehler werden nicht zuverlässig erkannt, es werden einfach neue lokale Variablen erzeugt.

```

to handle ..
....
system globVar -- deklariert eine globale Variable
....
lokVar = .. -- lokale Variable deklarieren
....
end

```

7.5.6 Kontrollstrukturen

Wie jede hohe Programmiersprache bietet **OpenScript** zahlreiche Kontrollstrukturen:

- *conditions/when/else* : Auswahl einer Option unter mehreren
- *if/then/else* : Programmverzweigung
- *while* : abweisende Schleife
- *do/until* : nichtabweisende Schleife
- *step* : Zählschleife, bei den meisten anderen Programmiersprachen als *for*-Schleife bekannt

7.5.7 Beispiel

In Bild 7.2 ist ein Ausschnitt aus einem komplexen Skript dargestellt.

7.6 weiterführende Möglichkeiten

Multimedia ToolBook ist so umfangreich, daß die Beschreibung aller Merkmale den Rahmen dieses Artikels sprengen würde. Deshalb stellt die folgende Liste einen kleinen Auszug aus den weiteren Möglichkeiten des Systems dar, ohne auf sie genauer einzugehen.

- Einbindung von VBX-Modulen (Module von Visual Basic)
- Schnittstelle zu DBase und Paradox
- Schnittstelle zum WWW
- OLE-Fähigkeit (nur als Client)
- Import von RTF-Texten (WinWord-Texte)
- Überblendeffekte

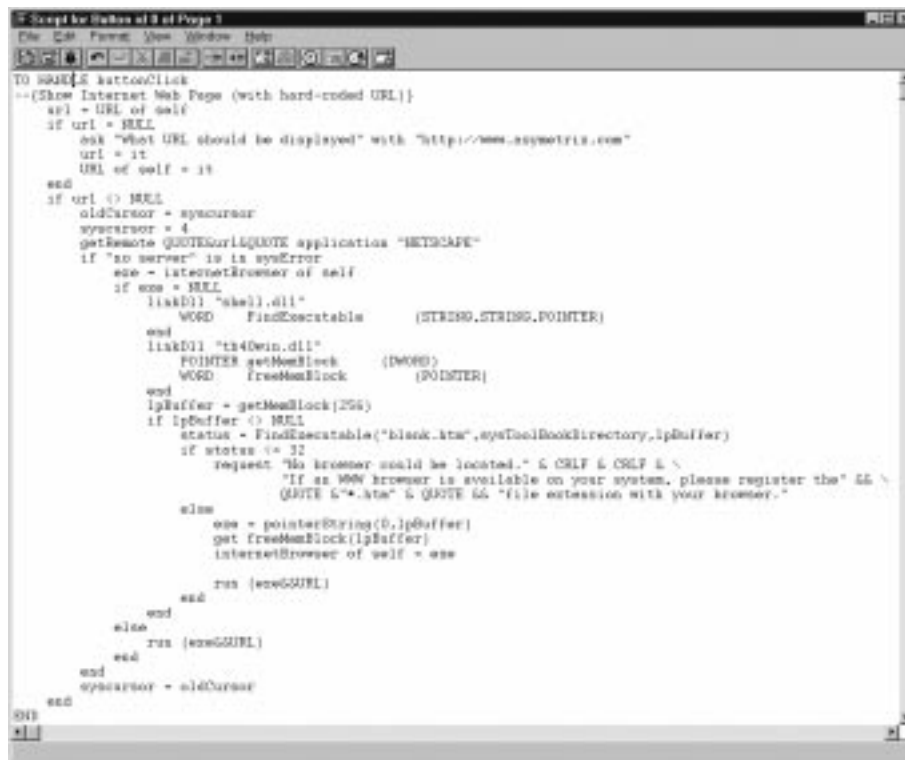


Abbildung 7.2: komplexes Skript

- Rechtschreibprüfung
- Paßwortschutz, man kann das Laden, das Speichern oder den Wechsel in den Autormodus durch Paßwörter schützen
- Script-Remover, entfernt alle Skripte aus einer fertigen Applikation, ohne die Funktionalität zu zerstören
- Speicherung der Dokumente als ausführbare Dateien (EXE-Dateien)

7.7 Verfügbarkeit

Das System ist nur unter Windows 3.11 / 95 / NT verfügbar. Eine Mac-Version wird von den Benutzern immer wieder verlangt, ist aber nicht in Sicht.

7.7.1 Preise

Asymetrix bietet das Autorensystem in der Form mehrerer verschiedener Editionen an. Außerdem wird das Programm **ToolBook** auch einzeln, ohne die multimediale Erweiterungen, angeboten. Die Tabelle 7.2 zeigt die Preise der einzelner Editionen.

7.7.2 Hardwareanforderungen

Durch den riesigen Umfang stellt das Autorensystem gewisse Anforderungen an die Hardware. Die Tabelle 7.3 zeigt die Anforderungen, die der Hersteller vorgibt, und die, mit denen eine ernsthafte

Editionen	Preise
Multimedia ToolBook 4.0	2100.00 DM
Multimedia ToolBook 4.0 CTB	2750.00 DM
Multimedia ToolBook 4.0 5er Lizenz	8400.00 DM
Multimedia ToolBook 4.0 CTB 5er Lizenz	11000.00 DM

Tabelle 7.2: Preise

Arbeit erst möglich wird. Alle Angaben sind Mindestgrößen.

	laut Hersteller	empfohlenes System
Prozessor	80386 SX / 20 MHz	Pentium 75
Hauptspeicher	8 MB	16
Festplatte	25 MB	150 MB
Graphik	640x480, 256 Farben	höhere Auflösungen, mehr Farben
Sonstiges	CD-ROM	schnelles CD-ROM
	Sound-Karte optional	gute Sound-Karte
	Maus	Maus

Tabelle 7.3: Hardware-Anforderungen

Das vom Hersteller angegebene System reicht höchstens für Windows 3.11 und eine kleine **ToolBook**-Applikation aus. Für Windows NT sollte man mindestens 32 MB RAM zur Verfügung haben.

7.8 Zusammenfassung

7.8.1 Vorteile

- einfache Bedienung
- einfaches, verständliches Konzept
- mächtige Skript-Sprache
- großer Lieferumfang
- zahlreiche Hilfen für Einsteiger
- Erweiterbarkeit

7.8.2 Nachteile

- geringe Performace des Systems
- keine 32-Bit-Merkmale bei Windows 95 / NT
- keine Unterstützung der Teamarbeit
- Wartung und Erweiterung bestehender Applikationen ist schwierig
- fehleranfälliges Variablenkonzept in **OpenScript**

7.8.3 Einsatzgebiete

Das Autorensystem kommt in den verschiedensten Gebieten des elektronischen Publizierens zum Einsatz:

- Gestaltung von CD-ROMs
- Präsentationen
- CBT
- Kiosk-Systeme
- Prototyping

7.8.4 Fazit

Das Autorensystem eignet sich hervorragend zum elektronischen Publizieren. Vor allem kleine und mittlere Applikationen können schnell und komfortabel erstellt werden. Große Anwendungen erfordern jedoch eine intensive Vorplanung, sonst kann die Übersicht recht schnell verlorengehen.

Die einfache Bedienung und das verständliche Konzept haben dazu geführt, daß dieses Autorensystem trotz der Einschränkung auf Windows-Plattformen bei vielen elektronischen Publikationen zum Einsatz kam.

7.9 Einige WWW-Adressen

<http://www.asymetrix.com/>
<http://weber.u.washington.edu/~brianp/>
<http://www.aviation.uiuc.edu/hpages/weller/virtcls/winauth.htm>
<http://www.biocom.arizona.edu/tbook/>

7.10 Literatur

- [Mana] *Multimedia ToolBook 3.0 User Manual.*
- [Manb] *OpenScript Reference Manual.*
- [Manc] *Multimedia ToolBook 4.0 Update.*
- [Mei94] Peter Meister. Rivalen der Fensterwelt. *Screen Multimedia*, Seiten 58–62, November 1994.
- [See96] Robert Seetzen. Kraftprotz. *c't*, Seiten 180–181, März 1996.
- [SR96] Thomas J. Schult und Dr. Peter Reimann. Virtuell lernen leicht gemacht. *c't*, Seiten 270–278, Januar 1996.
- [TB696] ToolBook Forum 6, September 1996.
- [TB796] ToolBook Forum 7, Dezember 1996.



Kapitel 8

APE Media Styler

Alexander Krieg

8.1 Übersicht

Bei dem APE Media Styler handelt es sich um ein Multimedia-Autorensystem, das laut Hersteller dem Autor die Erstellung einer Anwendung völlig ohne Programmierung erlaubt.

In diesem Kapitel wird zuerst das grundlegende Erstellungskonzept des Autorensystems behandelt. Danach wird eine Übersicht über die einzelnen Softwarekomponenten gegeben. Zum Schluß werden die Chancen, aber auch die Probleme, die sich bei diesem Erstellungskonzept ergeben, erörtert.

8.2 Einleitung

Die Anzahl der Multimedia-Anwendungen ist in den letzten Jahren rapide gewachsen. Die Gründe dafür sind hinlänglich bekannt. Ständig sinkende Hardwarepreise haben den PC und das Medium CD-ROM zum Massenartikel und den Multimedia-Markt zum Massenmarkt gemacht. Jedoch nicht nur die Schar der Konsumenten, sondern auch die der Autoren wird größer. Die Softwareindustrie reagiert ihrerseits auf diese Entwicklung mit immer umfangreicheren Werkzeugen, die den Autoren die Erstellung von Multimedia-Anwendungen erleichtern soll.

Das Softwareprodukt *APE Media Styler* ist ein Autorensystem, das ohne Programmiersprache auskommt. Damit soll insbesondere Autoren ohne Informatikkenntnisse der Weg von der Idee zur fertigen Multimedia-Anwendung erleichtert werden. Das aus dieser Zielsetzung resultierende Konzept und dessen Bewertung bilden den inhaltlichen Schwerpunkt dieses Kapitels.

8.3 Leistungsumfang

8.3.1 Die Versionen

Das Autorensystem *APE Media Styler* ist ausschliesslich auf der MS-Windows (zum gegenwärtigen Zeitpunkt nur Version 3.11) Plattform verfügbar und ist derzeit in 3 verschiedenen Versionen erhältlich:

Standard Version: Sie beinhaltet das komplette Autorensystem für die Entwicklung einer Multimedia-Anwendung.

Preis: 695,-DM [APE96]

Professional Edition: Sie beinhaltet das komplette Autorensystem, ein *Runtime-Modul* sowie eine Lizenz, die dem Autor erlaubt seine Anwendung kommerziell zu publizieren. Durch das *Runtime-Modul* kann die Multimedia-Anwendung selbstständig als eigene Anwendung ablaufen. Der Erwerb der *Professional Edition* ist somit für das Publizieren der Anwendung notwendige Voraussetzung.

Preis: 1995,-DM [APE96]

Special Edition: Sie beinhaltet das Autorensystem mit kompletter Funktionalität, aber mit limitierter Speichermöglichkeit. Durch diese Version können sich potentielle Käufer einen guten Eindruck über die Möglichkeiten, die die Software bietet, verschaffen.

Preis: 77,-DM [APE96]

8.3.2 Unterstützte Datenformate

Meistens liegen bei der Erstellung einer Multimedia-Anwendung eine Vielzahl der zu verarbeitenden Dateien bereits vor. Diese Dateien können sowohl aus dem Bereich Multimedia (Grafikdateien, Videodateien, Sounddateien), als auch aus anderen Bereichen (Textdateien, Datenbankdateien) stammen. Ein Autorensystem sollte in der Lage sein, möglichst viele verschiedene Datenformate aus den unterschiedlichsten Anwendungen zu verarbeiten, damit der Autor bei der Zusammenstellung seiner Quellen nicht unnötig eingeschränkt ist. Der Media Styler unterstützt die in Tabelle 8.1 angegebenen Datenformate.

Anwendungsbereich	Dateiformate
Grafik	JPEG, BMP, TGA, PCX EPS, GIF, TIF
Video	Apple Quicktime Video for Windows 8-Bit MPEG
Animation	Autodesk Animator Pro 3D Studio (FLI/FLC)
Audio	MIDI WAV
Datenbank	Dbase

Abbildung 8.1: Unterstützte Dateiformate

8.3.3 Unterstützte Hardware

Oftmals benötigen Multimedia-Anwendungen externe Geräte (z.B. Modem, Scanner) und Spezialhardware (z.B. Videokarten). Für den Autor wäre hierbei wünschenswert, daß er sich bei deren Einbindung nicht mit Gerätedetails auseinandersetzen müßte. Das Autorensystem sollte hier für die verschiedenen Hardwarekomponenten bereits vorkonfigurierte Schnittstellen bieten. Tabelle 8.2 zeigt, welche Hardware vom Media Styler ohne manuelle Konfiguration benutzt werden kann:

Hardware	Typ
Scanner	twainkompatibel
Videokarten	MPEG1 Motion JPEG 24-Bit
Laserdiskplayer	Pioneer Sony

Abbildung 8.2: Unterstützte Hardware

Die Einbindung der Multimedia-Hardware erfolgt über die Konfiguration der MCI Schnittstelle (*media control interface*), die die meisten Multimedia-Gerätetreiber unter MS-Windows verwenden. Zur Benutzung von Geräten an der seriellen Schnittstelle existiert ebenfalls ein Konfigurationsmenü.

8.3.4 Übersicht über die Schnittstellen

Abbildung 8.3 zeigt, welche Schnittstellen der APE Media Styler für die Einbindung externer Datenquellen anbietet.

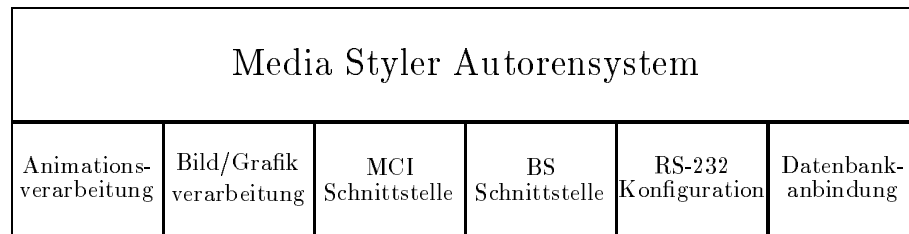


Abbildung 8.3: Übersicht über die Schnittstellen

8.4 Benutzungskonzept

8.4.1 Seitenbasierte Autorensysteme

Bei den Autorensystemen unterscheidet man bzgl. ihres Erstellungskonzeptes zwischen *seiten-*, *zeitachsen-* und *flußdiagrammbasierten* Ansätzen [Bol95]. Der Media Styler verfolgt den *seitenbasierten* Ansatz. Multimedia-Objekte werden hier auf Seiten (*Frames*) angeordnet. Eine Seite repräsentiert jeweils einen Benutzerbildschirm. Diese werden dem Benutzer in einer bestimmten Reihenfolge gezeigt. Bei den Multimedia-Objekten unterscheidet man zwischen *Informations-* und *Interaktionsobjekten*. Erstere dienen zur akustischen und visuellen Darstellung von Informationen. Letztere sind für die Interaktion mit dem Benutzer zuständig. Dieser kann über die *Interaktionsobjekte* nicht nur die Reihenfolge der Seiten beeinflussen, sondern auch Veränderungen an den *Multimedia-Objekten* selbst herbeiführen.

8.4.2 Benutzerführung

Der in Abschnitt 8.4.1 beschriebene seitenbasierte Ansatz zur Erstellung multimedialer Anwendungen läßt sich sehr gut in eine graphische Benutzeroberfläche wie MS-Windows integrieren. Die einzelnen *Seiten* werden durch *Fenster* repräsentiert. Mit einem *Toolkit* können die Multimedia-Objekte ausgewählt und danach mit der Maus per drag-and-drop in das entsprechende Fenster positioniert werden. Die Objekte können, wie man es beim Umgang mit Fenstern gewohnt ist, mit der Maus beliebig plaziert und formiert werden. Mit einem Doppelklick auf das entsprechende Objekt kann der Autor dessen Attribute editieren. Grundsätzlich wird dem Autor, soweit es möglich ist, bei allen Eingaben eine Auswahl von Optionen präsentiert. Tastatureingaben sind nur bei Namens- und bestimmten Parametereintragungen notwendig. Alle Einstellungen und Änderungen an den Objekten treten sofort in Kraft und sind dann weiter aktiv. Dadurch arbeitet der Autor unter einer vollständigen *WYSIWYG-Umgebung* in der es keine Trennung zwischen *Layout-*, *Strukturierungs-* und *Testphase* gibt.

8.5 Die Komponenten einer Media Styler Applikation

8.5.1 Übersicht

Tabelle 8.4 zeigt, wie die vorher auf abstrakter Ebene behandelten Komponenten im APE Media Styler realisiert wurden.

Multimediakomponente	Media Styler Komponente
Seite	Fenster (<i>window</i>)
Interaktionsobjekt	Auslöseknopf (<i>pushbutton</i>)
Informationsobjekt	Feldobjekt (<i>fieldobject</i>) Medienbox (<i>mediabox</i>) Textobjekt (<i>textobject</i>)

Abbildung 8.4: Repräsentation der Multimediakomponenten im APE Media Styler

8.5.1.1 Gemeinsame Attribute aller Multimedia-Objekte

ID-Nummer: Jedes Objekt bekommt eine eindeutige Nummer zur Identifizierung. Diese kann vom Autor nicht verändert werden.

Name (*name*): Die Namen der Objekte werden vom Programm automatisch generiert, jedoch sollte der Autor den Objekten eigene Namen geben, damit er sie besser identifizieren kann.

Farbe (*color*): Mit Hilfe eines Farbauswahlmenüs kann der Autor dem Objekt eine Farbe zuordnen.

Zeichensatz (*font*): Ein Zeichensatz wird über die Attribute *Typ*, *Stil*, *Größe* und *Farbe* vom Autor ausgewählt.

Schattentiefe (*shadow depth*): Legt die Tiefe des Schatteneffektes, mit dem die Schriftzeichen eines Objektes unterlegt werden können, fest.

Objektdarstellungspriorität: Da sich Objekte, die sich auf einem Fenster befinden überlappen können, muß festgelegt werden, welches Objekt über dem anderen liegt. Für jedes Fenster wird vom Autorensystem eine Liste, die die Darstellungsreihenfolge bestimmt, geführt. Die in der Liste definierte Reihenfolge kann in jeder Objektdialogbox vom Autor verändert werden.

Datenbankanbindung: Jedes Objekt kann Teil eines Datensatzes in einer Datenbank sein. Die Einbindung erfolgt über dieses Attribut.

Tabelle 8.6 zeigt die Dialogbox zum Einstellen der Objektattribute bei einem Auslöseknopf.

8.5.2 Fenster (*windows*)

8.5.2.1 Aufgaben

Ein Fenster repräsentiert jeweils einen Benutzerbildschirm. Jedes Multimedia-Objekt muß einem Fenster zugeordnet werden. Dia-Shows (*slide-shows*) werden durch Festlegen einer Fensterablauffolge definiert. Weitere wichtige Aufgaben der Fenster sind die Zugangskontrolle der Benutzer und die Anbindung einer Datenbank.

8.5.2.2 Wichtige Attribute

Anfangsaktion (*startbutton*) : Nach der Einblendung eines Fensters kann automatisch ein *Auslöseknopf* aktiviert werden.

Bild (*image*) : Jedes Fenster kann ein Hintergrundbild enthalten. Das Bild kann auf Wunsch der Bildschirmgröße angepaßt werden. (*stretch image*)

Sicherheitsstufe (*security level*) : Dem Autor steht beim Media Styler eine Benutzerverwaltung zur Verfügung. Die Benutzer, die sich beim Eintritt in die Multimedia-Anwendung mit einem Benutzernamen und Passwort identifizieren müssen, bekommen vom Autor eine Sicherheitsstufe zugeteilt. Hat ein Fenster eine höhere Sicherheitsstufe als der Benutzer, wird diesem der Zugang zum diesem Fenster verwehrt. Damit läßt sich auf einfache Weise eine Zugangskontrolle realisieren.

Einblendeeffekt (*fading*): Der Autor kann zwischen 100 verschiedenen Effekten zur Einblendung eines Bildschirms wählen. Die selben Effekte stehen auch für die Einblendung von Objekten zur Verfügung.

Datenbank (*database*) : Der Autor kann hier jedem Fenster eine Datenbank zuordnen.

8.5.3 Auslöseknöpfe (*pushbuttons*)

8.5.3.1 Aufgaben

Der Auslöseknopf spielt bei der Erstellung einer Media Styler Applikation eine zentrale Rolle. Zum einen ist er das einzige *Interaktionsobjekt*, das der Media Styler zur Verfügung stellt und somit die zentrale Schnittstelle für die Interaktion mit dem Benutzer. Zum anderen werden sämtliche dynamische Vorgänge der Multimedia-Anwendung durch Auslöseknöpfe initiiert. Ein solcher Vorgang besteht aus einer Aktionssequenz, die innerhalb eines Auslöseknopfes definiert wird. Dies bedeutet, dass erst durch die Auslöseknöpfe Beziehungen zwischen den *Multimedia-Objekten* entstehen. In den Auslöseknöpfen steckt demnach die gesamte Infrastruktur der Multimedia-Anwendung.

Ein Auslöseknopf wird in der Regel von einem Benutzer mittels der Maustaste aktiviert. Die Aktivierung kann allerdings auch durch eine Aktion innerhalb eines anderen Auslöseknopfes erfolgen. Da Auslöseknöpfe auch als unsichtbar definiert werden können, kann der Autor sie auch als kleine Programmmodule verwenden.

8.5.3.2 Aktionenkonzept

Wie bereits erwähnt, werden Veränderungen während des Ablaufs der Media Styler Applikation durch Aktionssequenzen herbeigeführt. Zur Definition einer Aktionssequenz wählt der Autor in der Aktionsdialogbox (siehe 8.7) eine aus 34 verschiedenen Aktionen aus, und ordnet sie danach in eine Liste, die die Ausführungsreihenfolge festlegt, ein. Die Parameter der Aktion werden über eine aktionsspezifische Dialogbox festgelegt. Tabelle 8.5 gibt eine Übersicht über alle zur Verfügung stehenden Aktionen..

8.5.3.3 Ausdrücke

Mit der Aktion *Data: Expression* kann ein Ausdruck berechnet und dessen Ergebnis in einem Feldobjekt gespeichert werden. Es existieren numerische, alphanumerische und logische Datentypen. Logische Ergebniswerte werden in einer Systemvariablen gespeichert und können für bedingte Aktionsausführungen benutzt werden. Als Operanden können Feldobjekte, Datenbankobjekte und Konstanten aller Datentypen verwendet werden. Als Operationen stehen logische Verknüpfungen, Relationen, Grundrechenarten sowie das Verketteten von Zeichenketten zur Verfügung. Weiterhin kann der Autor auf Funktionen, die bereits aus Datenbankprogrammen bekannt sind, zurückgreifen. (Siehe 8.8)

8.5.4 Textobjekte (*text object*)

Das Textobjekt dient zur Eingabe und zum Anzeigen von Zeichenketten. Es können externe Textdateien (reiner ASCII-Code) eingebunden werden. Durch einfaches Anklicken des entsprechenden Textobjektes kann der Benutzer, sofern dies vom Autor eingestellt wurde, ein Textobjekt editieren. Dabei wird dem Benutzer ein horizontaler und vertikaler Rollbalken zur Navigation im Text zur Verfügung gestellt.

8.5.5 Medienbox (*media box*)

Die Medienbox dient zur Ausgabe von statischer und dynamischer Grafik. Die Grafikausgabe wird dabei den Ausmaßen der Medienbox angepaßt. In der Medienbox können Videos und Animationen abgespielt werden. Aber auch normale Bilddateien werden über die Medienbox angezeigt.

8.5.6 Feldobjekt (*field object*)

Das Feldobjekt dient als universeller Wertebehälter. Dabei kann es im Benutzerkontext als Eingabefeld, im Aktionenkontext als Variable und im Datenbankkontext als Datenbankfeld benutzt werden.

Wirkungsbereich	Aktionsname	Beschreibung
Daten	Check Expression Message Read Serial Port	Feldobjekte auf korrekte Antwort überprüfen Ausdruck berechnen und in Objekt speichern Messagebox erstellen, Lese Daten von Datei Lese Daten von serieller Schnittstelle
Datenbank	Add Record Close & Open Delete Goto Record Index & Filter	Datensatz zur Datenbank hinzufügen Schliessen und Öffnen von Datenbanken Löschen eines Datensatzes Springe zu Datensatz Filterfunktionen auf Datensätze verwenden
MCI	Movie Sound	Videodatei abspielen Sounddatei abspielen
Objekt	Delete Do Path Field Edit Image Modify	Objekt löschen Objekt entlang eines Pfades bewegen Benutzer kann Feldobjekt editieren Objekt neues Bild zuordnen Verändern von Objektattributen
Programm	DOS Execute Quit Return Save	Aufrufen eines MS-DOS Programms Aufrufen einer Windows-Anwendung Media Styler Applikation verlassen zurück zur aufrufenden Applikation springen Speichern von Veränderungen an der Applikation
Bildschirm	Define Path Menu Refresh Slide	Bewegungspfad definieren User-Pop-Up-Menü aktivieren Bildschirminhalt aktualisieren Dia-Show aktivieren
Schritt	Check Box Delay Do Button Do Nothing Radio Button Terminate	Dialogbox aktivieren Ausführung der nächsten Aktion verzögern Auslöseknopf aktivieren Nichts tun (nützlich bei Verzweigungen) Aktivieren einer Dialogbox Ausführung einer Aktionensequenz abbrechen
Fenster	Device Toolkit	Konfigurieren der seriellen Schnittstelle Benutzertoolkit aktivieren

Abbildung 8.5: Übersicht über die Aktionen

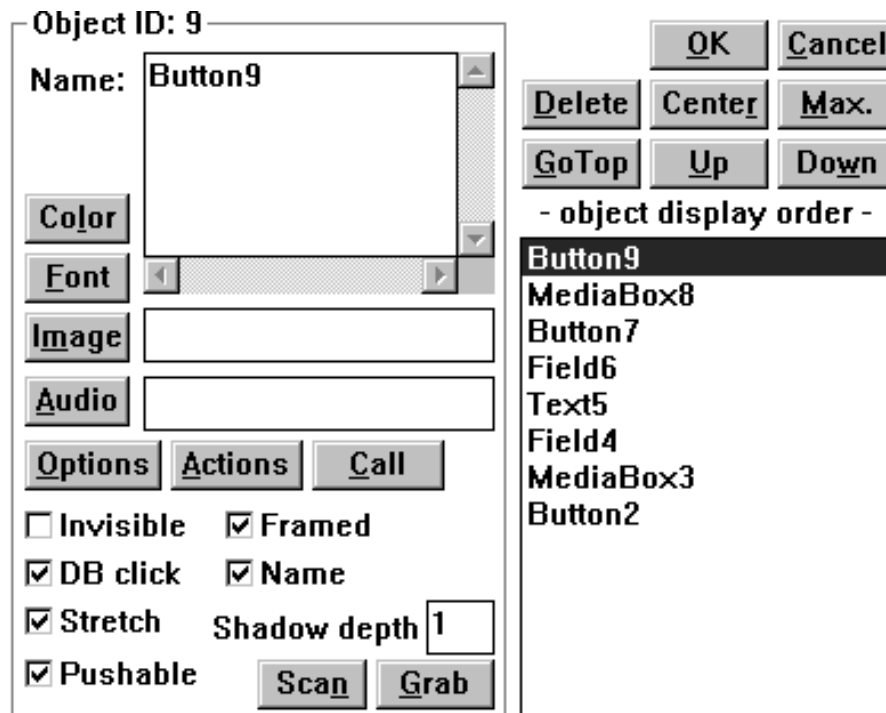


Abbildung 8.6: Dialogbox der Objektattribute (Auslöseknopf)

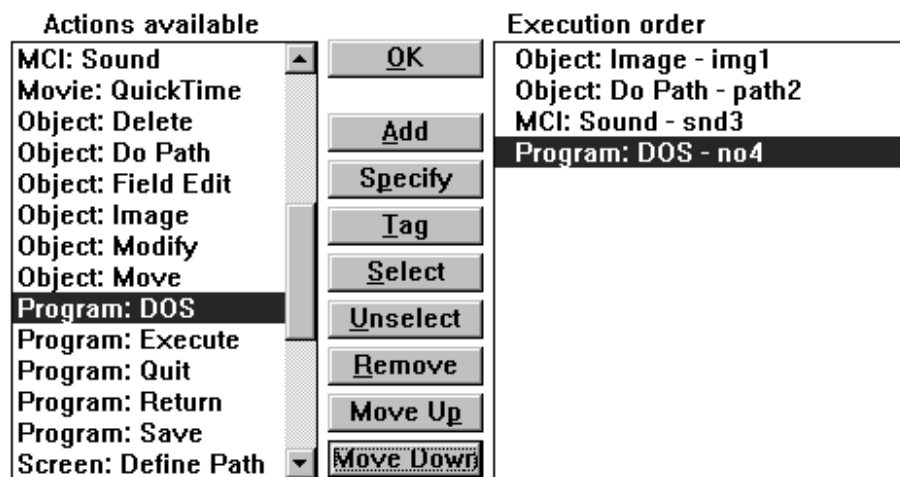


Abbildung 8.7: Aktionendialogbox

Art	Funktion	Beschreibung
Konvertierungen	CTOD(String)	Konvertiert Zeichenkette in Datum
	DTOC(Datum)	Konvertiert Datum in Zeichenkette
	DTOS(Datum)	wie DTOC nur anderes Format
	STOD(String)	Konvertiert Zeichenkette in numerischen Wert
	STR(W,L,D)	Konvertiert numerischen Wert W in Zeichenkette mit Länge L und D Dezimalstellen
	VAL(String)	Konvertiert Zeichenkette in numerischen Wert
Zeichenkettenmanipulationen	UPPER(String)	Wandelt alle Zeichen in Großbuchstaben um
	TRIM(String)	Leerzeichen am Ende einer Zeichenkette werden abgeschnitten
	LTRIM(String)	Leerzeichen am Anfang einer Zeichenkette werden abgeschnitten
	SUBSTR(String,P,L)	Schneidet an der Position P eine Zeichenkette der Länge L aus
Systemfunktionen	DATE()	Gibt aktuelles Datum aus
	TIME()	Gibt aktuelle Systemzeit aus
	DAY(Date)	Gibt den Wochentag eines Datums aus
	MONTH(Date)	Gibt den Monat eines Datums aus
	YEAR(Date)	Gibt das Jahr eines Datums aus
Datenbank	DELETE()	Prüft, ob ein Datensatz zum löschen markiert ist
	RECCOUNT()	Gibt Anzahl der Datensätze in einer Datenbank an
	RECNO()	Gibt die aktuelle Datensatznummer aus
Struktur	IIF(L,TrVal,FalVal)	Ist der logische Ausdruck L wahr, wird der Wert TrVal zurückgegeben ansonsten wird der Wert FalVal zurückgegeben

Abbildung 8.8: Funktionen innerhalb eines Ausdrucks

8.5.6.1 Datenbankbindung

Der Media Styler besitzt eine umfangreiche Schnittstelle zu Datenbanken. Der Einsatz der Datenbank kann in zwei Richtungen erfolgen. Einerseits kann der große Informationsgehalt, der in bereits bestehenden Datenbanken vorhanden ist, von der Media Styler Applikation genutzt werden. Andererseits können zur Archivierung von Benutzereingaben auch neue Datenbanken erstellt werden. In erster Linie dienen die Feldobjekte als Datenbankfelder, aber auch alle anderen Multimedia-Objekte können in die Datenbank integriert werden. Für die Auswahl von Datensätzen können Filter verwendet werden. Zur Erstellung eines Filterausdrucks stehen logische Operatoren und Relationen zur Verfügung.

8.6 Komponenten zur Erstellung einer Media Styler Applikation

8.6.1 Toolkit

Das Toolkit ist ein kleines Fenster, dessen Inhalt aus 24 Icons besteht. Die Icons sind in zwei Hälften unterteilt, wobei immer nur eine Hälfte im Fenster sichtbar ist. Durch einfaches Anklicken der Icons erhält der Autor einen schnellen Zugriff auf oft benötigte Funktionen des Autorensystems. Die Toolbox kann aber auch Benutzern der fertigen Media Styler Applikation zur Verfügung gestellt werden. Dann umfasst sie allerdings nur die zweite Hälfte der Funktionen, welche keinen Eingriff in die innere Struktur der Applikation erlauben.

8.6.2 Bildverarbeitung (*image processing*)

Zur optischen Gestaltung einer Multimedia-Anwendung ist das Verwenden von Bildern unerlässlich. Der Media Styler hat dazu eine Bildverarbeitung als festen Bestandteil des Autorensystems integriert. Die Grafikdaten können aus Dateien oder direkt von einem Scanner gelesen werden. Die Bearbeitungsfunktionen können zwar ein professionelles Bildverarbeitungsprogramm nicht vollständig ersetzen, jedoch kann der Autor eine Vielzahl der Routinearbeiten an den Grafiken innerhalb des Autorensystems erledigen.

8.6.3 Bewegungspfad für Objekte (*move-path feature*)

Um eine Multimedia-Anwendung interessanter zu gestalten, werden oft Animationen eingesetzt. Das flüssige Bewegen von Objekten wird im Media Styler über den Einsatz von Bewegungspfaden erreicht. Nach Auswahl der entsprechenden Funktionen wird der Bewegungspfad einfach mit der Maus (bei gedrückter Taste) über das aktuelle Fenster gemalt. Danach wird der Bewegungspfad in eine eigene Datei gespeichert. Mittels der *Do-Path-Aktion* können Objekte diesen Pfad entlang mit einstellbaren Geschwindigkeiten bewegt werden.

8.6.4 Dia-Show-Editor (*slide show editor*)

Eine Dia-show ist eine vordefinierte Ablauffolge von Bildern. Im Dia-show Editor kann der Autor eine solche Ablauffolge durch eine Liste von Fenstern definieren. In dieser Liste wird neben dem Fensternamen auch noch die Dauer der Einblendung des Fensters sowie eine Angabe, ob die Startaktion des Fensters ausgeführt werden soll, eingetragen. Die fertige Liste wird als Slide-Show-Datei abgespeichert. Mit der Aktion *Slide* kann durch Angabe einer solchen Datei eine Diashow in der Anwendung ablaufen.

8.6.5 Benutzer Pop-Up-Menü (*user pop-up menu*)

Ein Pop-Up-Menü ist ein Menü, das unmittelbar auf Tastendruck an der aktuellen Mauszeigerposition erscheint. Innerhalb der Media Styler Applikation dient das Pop-Up-Menü dem Benutzer ausschließlich zum Aufrufen neuer Bildschirmseiten. Zur Definition eines solchen Menüs bietet der Media Styler dem Autor eine spezielle Dialogbox an. Im Dialogfeld gibt der Autor die entsprechenden Fensternamen,

sowie eine Sicherheitsstufe an. Zur besseren Strukturierung des Pop-Up-Menüs werden die Fensternamen in Untermenüs angeordnet. Als weitere Gestaltungsmittel stehen Trennungslinien (*separator*) und die Definition von Tastaturkürzeln (*hotkeys*) zur Verfügung. Für eine Media Styler Applikation kann nur ein universelles Pop-Up-Menü definiert werden. Eine Realisierung von kontextsensitiven Menüs, die heutzutage in vielen Softwareanwendungen implementiert sind, ist nicht möglich.

8.7 Bewertung

8.7.1 Vorteile des Konzepts

Der Mensch-Maschine-Dialog im Media Styler ist konsequent menügesteuert. Die Auswahl der Optionen findet in Dialogboxen statt, dessen Bedienung bereits aus der MS Windows-Umgebung bekannt ist. Der Autor arbeitet so in einer Entwicklungsumgebung mit der er in der Regel bereits vertraut ist. Durch den Verzicht einer Skriptsprache muß der Autor weder Semantik noch Syntax einer solchen erlernen. Durch das WYSIWYG-Konzept sieht der Autor während der gesamten Entwicklung die Anwendung aus dem Blickwinkel des späteren Benutzers. Somit benötigt der Autor bei der Entwicklung wenig Abstraktionsvermögen und kann Fehlentwicklungen unmittelbar erkennen.

Zum Testen der Anwendung sind keine Voreinstellungen nötig. Vielmehr findet ein erster Test bereits implizit während der Erstellung statt. Dadurch kommt man sehr schnell zu ersten Ergebnissen, die sich auf Autor und Auftraggeber motivierend auswirken können.

Da keine Trennung der Arbeitsschritte „Layouten“ und „Strukturieren“ vorgesehen ist, hat der Autor im Entwicklungsprozeß freie Wahl über die Reihenfolge seiner Arbeitsschritte. So kann er sich zum Beispiel zuerst um die optische Gestaltung der Anwendung kümmern (*Layoutphase*) und danach die Beziehungen zwischen den Objekten definieren (*Strukturierungsphase*). Ebenso kann der Entwicklungsprozeß in umgekehrter Reihenfolge durchgeführt werden. Oftmals wird der Autor jedoch versuchen, beide Arbeitsschritte gleichzeitig zu bewältigen, d.h. er wird sich bei der Gestaltung eines neuen Multimedia-Objektes auch gleich um dessen Eingliederung in die Infrastruktur der Anwendung bemühen. Auf diese Weise läßt sich die Anwendung sukzessiv bis zum gewünschten Endstadium erweitern.

Die wesentlichen Vorteile des Konzeptes im APE Media Styler lassen sich also folgendermaßen zusammenfassen:

- + Umgang mit dem Autorensystem fällt sehr leicht
- + sehr schnelle Erstellung von Prototypen
- + problemloses Erweitern der Anwendung

8.7.2 Nachteile des Konzepts

Wie in Abschnitt 8.5.3.1 beschrieben, werden Beziehungen zwischen den Multimedia-Objekten mit Aktionen, die jeweils in den innerhalb eines *Auslöseknopfes* definiert sind, hergestellt. Die Informationen über diese Beziehungen erhält man in den Aktions-Untermenüs der Auslöseknöpfe. Dabei hat der Autor jedoch immer nur Einblick auf Vorgänge, die von einem einzelnen Auslöseknopf initiiert werden. Es fehlt eine Übersicht über die gesamte Infrastruktur der Applikation. Durch den fehlenden Blick auf die Gesamtheit der Zusammenhänge, wird es, mit zunehmender Komplexität der Applikation, für den Autor immer schwieriger ihren gesamten Ablauf zu erfassen. Hinsichtlich der *Wartbarkeit* von Applikationen wirkt sich dieser Nachteil besonders gravierend aus. Auch *Teamarbeit* läßt sich unter diesem Konzept nur schwer realisieren.

Die verschiedenen Aktionen bieten dem Autor eine große Vielfalt von Manipulationsmöglichkeiten an den Objekten. Jedoch sind die Anwendungsmöglichkeiten der Aktionen mitunter sehr unflexibel. So sind beispielsweise keine Variablen als Parameter einer Aktion zugelassen. Mit dem Aktionenkonzept lassen sich auch nur sehr einfache Programmstrukturen realisieren. Bei der Umsetzung von komplexeren Algorithmen muß der Autor auf externe Programme zurückgreifen.

Das Benutzungskonzept legt dem Autor ein konzeptloses Vorgehen bei der Entwicklung seiner Applikation nahe. Dies mag für den Autor zwar auf den ersten Blick sehr attraktiv erscheinen, kann aber in der Praxis häufig dazu führen, daß er die Orientierung verliert und sich in seiner eigenen Anwendung nicht mehr zurecht findet.

Bei einem Verzicht auf eine Skriptsprache fallen natürlich auch deren Vorteile weg. So kann man für immer wiederkehrende Strukturen keine Bibliotheken oder Templates anlegen.

Die Nachteile lassen sich folgendermaßen zusammenfassen:

- keine Übersicht über die Infrastruktur
- Aktionskonzept unflexibel
- konzeptloses Vorgehen wird unterstützt
- schlechte Wiederverwendbarkeit

8.7.3 Arbeiten in der Praxis

In der Praxis erweist sich das Arbeiten mit dem Media Styler als sehr angenehm. Das Ziel, einen schnelleren Zyklus zwischen Layout-, Strukturierungs- und Testphase zu schaffen, wurde gut umgesetzt. Leider bietet die diesem Kapitel zugrunde liegende Version des Media Stylers keine Undo-Funktion. Dies macht sich insbesondere deshalb negativ bemerkbar, weil viele destruktive Funktionen ohne Bestätigungsabfrage unmittelbar ausgeführt werden.

8.7.4 Fazit

Mit dem Autorensystem *APE Media Styler* lassen sich einfache Multimedia-Anwendungen sehr schnell auch von Autoren ohne Programmierkenntnisse erstellen. Insbesondere für die Gestaltung einer graphischen Oberfläche zur Darstellung von multimedialen Daten ist das Autorensystem hervorragend geeignet. Auch die Möglichkeiten zur Nutzung von Datenbanken sind sehr umfangreich und geben dem Autor einen großen Handlungsspielraum. Für die Realisation umfangreicherer Multimedia-Anwendungen ist eine Aufteilung in kleine Teilanwendungen, die sich gegenseitig aufrufen, unbedingt erforderlich. Die Entwicklung von Anwendungen, die Algorithmen enthalten, erweist sich als schwierig oder gar unmöglich. Hier bleibt dem Autor nur die Möglichkeit externe Programme zur Lösung der Aufgabe einzubeziehen.

Vor dem Hintergrund, daß Multimedia immer mehr im Heim- und Hobbybereich zur Anwendung kommt, hat ein Autorensystem mit diesen Eigenschaften sicherlich sehr gute Zukunftsperspektiven. Obwohl der Preis im Vergleich mit anderen Autorensystemen sehr positiv ins Gewicht fällt, sollte man, im Hinblick auf die neue Autorenzielgruppe, von Seiten der Entwickler nochmals über die Preisgestaltung des Produkts nachdenken.

8.8 Literatur

[APE96] APE. Multimedia Preisliste. Papier, Mai 1996.

[Bol95] Dietrich Boles. Elektronisches Publizieren. In W. Neubauer, Hrsg., *Deutscher Dokumentartag 1995*, Seiten 393–411. Deutsche Gesellschaft für Dokumentation, URL: <http://www-is.informatik.uni-oldenburg.de/~dibo>, Sept. 1995.



Kapitel 9

Authorware

Andreas Steinbuch

9.1 Übersicht

In diesem Kapitel wird ein Überblick über das Autorensystem Authorware von Macromedia, Inc. gegeben.

9.1.1 Einleitung

Der Multimedia-Markt boomt. Waren Anfang 1995 über 11.000 Multimedia-Titel weltweit verfügbar, so waren es zu Jahresbeginn 1996 schon 16.000. Dies ist ein Zuwachs von über vierzig Prozent. Bis zum Jahr 2000 rechnet das Markforschungsinstitut Infotech mit ca. 40.000 verfügbaren Titeln [mul96].

Auch auf dem deutschen Markt kündigen sich für den Multimedia-Bereich rosige Zeiten an. 1995 sind mehr als doppelt so viele deutschsprachige Multimedia-Publikationen erschienen wie im Vorjahr, dabei wurden im zweiten Halbjahr 1995 doppelt so viele neue Titel wie im ersten Halbjahr veröffentlicht. Kamen von Januar bis Juni durchschnittlich rund 50 Titel pro Monat neu in die Verkaufsregale, steigert sich die Zahl im zweiten Halbjahr auf durchschnittlich 95 pro Monat [mul95a].

Deutschland wird zum umsatzstärksten Markt in Europa. Allein im Bereich der Lernsoftware wird der deutsche Markt von gegenwärtig 86 Millionen auf knapp 1,3 Milliarden Mark anwachsen. Bis zum Jahr 2005, so prognostiziert die englische Marktforschungsgesellschaft Datamonitor, wird der private Edutainment-Markt in Deutschland gar 3,3 Milliarden umsetzen [mul95b].

Der amerikanische Multimedia-Spezialist Macromedia entwickelt die notwendigen Werkzeuge zur Erstellung von multimedialen Anwendungen. Er stellt dem Entwickler eine ganze Palette von Produkten zum Entwickeln und Bearbeiten von Graphik-, Audio- und Video-Informationen zur Verfügung. Das Autorensystem Authorware, Gegenstand dieser Abhandlung, ist laut Herstelleraussagen die mächtigste Entwicklungsumgebung zur Erstellung von interaktiven Informationssystemen [Inf96].

9.1.2 Einordnung

Abstrakt gesehen sind multimediale Anwendungen spezielle Programmsysteme, die sich mit Hilfe moderner Programmiersprachen implementieren lassen. Voraussetzungen hierfür sind ein nicht zu unterschätzender Zeit- und Finanzbedarf sowie ausgereifte Programmiererfahrung.

Im Widerspruch hierzu stehen die Entwickler multimedialer Applikationen. Nur die wenigsten haben Programmiererfahrung oder verfügen über eine fundierte Informatikausbildung. Meistens handelt es sich hier um Designer, Werbefachleute oder Ausbilder. Diesem Personenkreis, aber nicht nur diesem, müssen höhere Softwarewerkzeuge zur Verfügung gestellt werden, die eine einfache und schnelle Entwicklung multimedialer Anwendungen ohne oder mit nur geringen Programmierkenntnissen unterstützen. Derartige Werkzeuge werden Autorensysteme genannt.

Sie helfen dem Entwickler bei der Aufbereitung der zu präsentierenden Informationen, indem sie ihn teilweise oder vollständig von der programmiertechnischen Realisierung befreien. Der Entwickler entwirft seine multimediale Präsentation, indem er Beziehungen zwischen Medienobjekten (Text, Audio, Video usw.) herstellt. Bezüglich der Art und Weise, wie zeitliche Beziehungen zwischen den Medienobjekten spezifizierbar sind, können existierende Autorensysteme in drei Klassen eingeteilt werden:

- seitenbasierte Autorensysteme,
- zeitbasierte Autorensysteme und
- Flußlinien-basierte Autorensysteme.

Bei seitenbasierten Autorensystemen erfolgt die Programmierung einer Anwendung durch das Anordnen der einzelnen Medienobjekte und Programm-Funktionen auf einer Seite. Diese Seiten werden zu einem Buch zusammengefaßt (Bsp.: ToolBook von Asymetrix). Werden die einzelnen Medienobjekte auf einer Zeitachse (Drehbuch) angelegt und mit logischen Programmfunktionen verknüpft, dann spricht man von einem zeitbasierten Autorensystem. Ein typischer Vertreter dieser Klasse ist Director von Macromedia. Das hier zu besprechende Autorensystem, Authorware von Macromedia, gehört zu der dritten Kategorie. Authorware arbeitet mit Icons, die die Medienobjekte repräsentieren. Sie werden in einem Flußdiagramm (engl. flowchart) angeordnet und durch Kanten miteinander verbunden. Die Kanten bestimmen den Programmablauf. Trifft das Programm auf eine Verzweigung, so muß eine Entscheidung getroffen werden, welcher Kante der Programmablauf folgen soll. Eine solche Entscheidung kann zum Beispiel nach einer Eingabe durch den Benutzer getroffen werden.

Durch dieses Verfahren lassen sich auch ohne das Erlernen einer Programmiersprache komplexe und anspruchsvolle multimediale Anwendungen realisieren. Um es gleich vorweg zu nehmen, auch bei Authorware wird man am Programmieren nicht herumkommen, will man das Potential von Authorware voll ausschöpfen. Die Entwickler haben bei Authorware jedoch darauf geachtet, daß die Notwendigkeit von Programmroutine auf ein Mindestmaß reduziert wurde und nur da verwendet wird, wo es absolut nötig ist.

9.2 Anwendungsgebiete für Authorware

Anfangs wurde Authorware speziell zum Entwurf von Lehrsoftware für Macintosh-Rechner entwickelt [DPR96]. Inzwischen erstrecken sich die Einsatzmöglichkeiten dieses Autorensystems für multimediale Applikationen über eine ganze Reihe von Anwendungsgebieten:

- Lehr- und Trainingssoftware zur Aus- und Weiterbildung (Courseware und CBT: Computer-Based-Training)
- Produkt- und Firmenpräsentationen
- Interaktive Publikationen, Dokumentationen und Kataloge
- Kiosksysteme/Infoterminals
- Simulationen und Prototypengenerierung
- Unterhaltung

Diese Verwendungsvielfalt, ermöglicht durch eine kontinuierliche Weiterentwicklung und Erweiterung, zeichnet Authorware als ein reifes und mächtiges Softwareprodukt aus. Die aktuelle Versionsnummer von Authorware wird von Macromedia mit 3.5.1 beziffert.

9.3 Plattformverfügbarkeit

Authorware wird von Macromedia für Windows-Betriebssysteme (Windows 3.1, 95 und NT), als auch für Macintosh-Rechner (Macintosh und PowerMacintosh) vertrieben. Der Hersteller empfiehlt für die Windows-Version einen mit 75 MHz getakteten Pentium-Rechner (WindowsNT mit 100 MHz) oder besser [Mac96b]. Zusätzlich sollte der Rechner mit mindestens 16 MB Hauptspeicher (WindowsNT mit 32 MB) und einer beschleunigten PCI-Graphikkarte ausgestattet sein. Authorware selbst benötigt 15 MB Plattenspeicher, jedoch wird für ein sinnvolles Arbeiten mindestens ein freier Speicherplatz von 30 MB für das System empfohlen. Das Produkt unterstützt die marktgängigen Soundkarten.

Für die Macintosh-Variante empfiehlt Macromedia einen 68040 mit 33 MHz oder einen PowerMacintosh mit mindestens 75 MHz. Auf dem Rechner muß das Betriebssystem System 7.1 oder höher installiert sein und ein Hauptspeicher von 12 MB zur Verfügung stehen. Auch hier sollte, wenn möglich, dem Programm 30 MB Plattenspeicher eingeräumt werden.

Mit Authorware erstellte Applikationen können von der einen Plattform auf die andere übertragen werden. Jedoch nur unter der Voraussetzung, daß der Anwender die Lizenz für beide Versionen erstanden hat.

9.4 Arbeitsoberfläche

Die beiden Versionen, Authorware für Windows-Rechner und Authorware für Macintosh und PowerMacintosh, unterscheiden sich in ihrer Funktionalität und in ihrem Aufbau nur minimal. Da beide Versionen dem jeweiligen Betriebssystem angepaßt wurden, ergeben sich geringe Unterschiede in der Oberflächenpräsentation und Benutzerführung (z.B. Command-Taste bei Macintosh statt Control-Taste). Der größte Unterschied besteht in der Unterstützung unterschiedlicher Dateiformate, die importiert und bearbeitet werden können. Naturgemäß werden bei der Windows-Variante eine Vielzahl mehr Formate unterstützt, als bei den Macintosh-Rechner [Mac96b]. Dies kann man nicht als Nachteil für die Mac-Version anführen, da bei diesem Computertyp mehr Wert auf einen einheitlichen Standard für Dateiformate gelegt wurde.

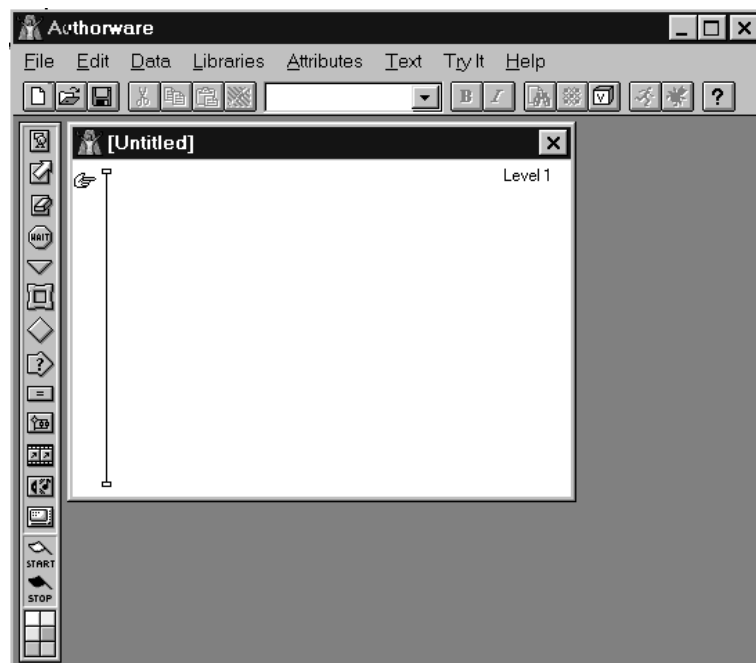


Abbildung 9.1: Arbeitsoberfläche auf Windows 95

Betrachtet man sich nach dem Starten des Programms das Authorware-Hauptfenster, so ist am auffälligsten eine Auswahlleiste mit 13 Icons, zwei Flaggen und einer kleinen Farbpalette. Sie befindet sich am linken Rand direkt unterhalb der Kopfleiste mit den bekannten Pull-Down-Menüs und einer weiteren Knopfleiste.

Die 13 Icons repräsentieren Objekte, die sich im Flußdiagramm einbinden lassen. Per Drag-und-Drop-Verfahren kann man die Icons aufnehmen und an gewünschter Stelle im Flußdiagramm platzieren. Die Kanten in diesem Diagramm bestimmen die möglichen Pfade für den Präsentationsablauf. Um nun einen genaueren Überblick über das Programm zu erlangen, sollen die Funktionalitäten der Icons näher betrachtet werden:

9.4.1 Darstellungs-Icon (display)

Eines der wichtigsten Funktionen erfüllt das Darstellungs-Icon, denn es ist für die Wiedergabe von Text und Graphiken aller Art zuständig. Wird es im Flußdiagramm mit einem Doppelklick aktiviert, dann wird das entsprechende Präsentationsfenster geöffnet. Zusätzlich öffnet sich eine Graphikbearbeitungsleiste, mit einfachen Funktionen wie z.B. Kreis, Rechteck, Linie, Polygon. Auch das Einfügen von Text ist hier möglich.

Über die Funktion Import im Pull-Down-Menü File lassen sich Graphiken einbinden. Diese Graphiken können dann im Präsentationsfenster nach Belieben platziert und darüber hinaus skaliert werden.

9.4.2 Bewegungs-Icon (motion)

Das Bewegungs-Icon erlaubt es, Objekte im Präsentationsfenster nach unterschiedlichen Regeln zweidimensional zu bewegen. So kann ein Objekt z.B. von jeder beliebigen zu einer fix definierten Position bewegt werden. Ebenso ist die Definition eines freien Pfades möglich. Die Geschwindigkeit der bewegten Objekte kann in zwei Varianten bestimmt werden. Erste Möglichkeit ist die Angabe in Sekunden für den ganzen Bewegungsprozeß, unabhängig von der Distanz. Als Alternative läßt sich aber auch die Geschwindigkeit bestimmen, indem man die Zeit angibt, die benötigt wird um, 72 Pixel (ca. ein Inch) zu überbrücken. Die Auswahl der Objekte, die sich animieren lassen, ist nicht auf Graphiken und Text beschränkt. Es können sich auch Bildsequenzen bewegt werden.

9.4.3 Lösche-Icon (erase)

Mit dem Lösche-Icon wird die Löschung von graphischen Objekten vom Bildschirm während der Präsentation definiert. Dabei werden die zu löschenden Objekte einfach über das Flußdiagramm bestimmt.

9.4.4 Warte-Icon (wait)

Mit dem Warte-Icon lassen sich Pausen in den Ablauf einbauen, die durch Tastendruck, Mausklick oder automatisch nach einer vorgegebenen Zeitspanne beendet werden.

9.4.5 Navigations-Icon (navigation)

Das Navigations-Icon erlaubt direkte Sprünge zu Icons, die in Verbindung zu einem Rahmen-Icon stehen. Ein Sprung kann automatisch vollzogen werden oder in Verbindung mit einem Interaktions-Icon. Hier kann der Benutzer zwischen mehreren Alternativen auswählen und bestimmt so das Sprungziel. Navigations-Icons brechen die starre Struktur eines regulären Flußdiagramms auf. Dies ermöglicht eine größere Gestaltungsfreiheit beim Ablauf, reduziert jedoch die Übersichtlichkeit im Flußdiagramm.

9.4.6 Rahmen-Icon (framework)

Mit einem Rahmen-Icon lassen sich verschiedene Präsentationssequenzen miteinander verknüpfen. Über Navigations-Knöpfe (vorwärts, zurück, letzte Seite, finden usw.), die bei der Erstellung des Rahmen definiert werden, ist es dem Benutzer möglich, auf einfache Weise durch die Sequenzen zu navigieren. Auch hier wird die starre Struktur eines Flußdiagramms verletzt, da Präsentationssequenzen übersprungen oder zwischen mehreren Abfolgen gewechselt werden kann.

9.4.7 Entscheidungs-Icon (decision)

Durch die Verwendung eines Entscheidungs-Icon lassen sich Pfadverzweigungen und Schleifen im Ablauf realisieren. Dabei kann die Entscheidung, welchem Pfad gefolgt werden soll, in Reihenfolge, zufällig oder durch Berechnung bestimmt werden. Wie oft das Entscheidungs-Icon durchlaufen werden soll, kann durch die Angabe der Wiederholungen oder durch eine Zeitbegrenzung bestimmt werden. Nach Beendigung wird wieder zum Hauptpfad, der nach dem Entscheidungs-Icon folgt, zurückgekehrt.

9.4.8 Interaktions-Icon (interaction)

Die Funktionalität hier ähnelt dem Entscheidungs-Icon. Wichtigster Unterschied ist, daß der Benutzer durch Mausklick auf einen Knopf oder ausgewählten Bereich (hotspot) entscheidet. Möglich ist aber auch die Eingabe eines Zeichens oder Textes, welcher Ablaufpfad eingeschlagen werden soll.

9.4.9 Berechnungs-Icon (calculation)

Mit diesem Icon lassen sich Skripte in das Flußdiagramm einbinden. Hierfür stehen If-Then-Konstrukte, Schleifen und Funktionen zur Verfügung, mit denen sich Befehlsabfolgen definieren lassen.

9.4.10 Gruppierungs-Icon (map)

Das Gruppierungs-Icon ermöglicht das Zusammenfassen mehrerer, aufeinander folgender Icons zu einem Icon. Dies dient zur Erhöhung der Übersichtlichkeit im Flußdiagramm. Mehrere Gruppierungs-Icon lassen sich auch wieder zu einem Icon zusammen fassen.

9.4.11 Digital-Movie-Icon (digital movie)

Durch die Verwendung dieses Icons lassen sich digitale Bildsequenzen importieren. Authorware ermöglicht die Wiedergabe nur eines bestimmten Teils, in dem das Anfangs- und das Endbild bestimmt wird. Des weiteren lassen sich die Abspielgeschwindigkeit, Transparenz, Interaktivität (nur bei Sequenzen, die mit dem Director von Macromedia erstellt wurden) und Anzahl der Wiederholungen bestimmen.

9.4.12 Ton-Icon (sound)

Mit dem Ton-Icon können Audioquellen in die Präsentation eingebunden werden. Die Wiedergabe kann einmalig oder wiederholt erfolgen. Interessant ist die direkte Verknüpfung mit einer Bedingung oder Variable. Sobald die Bedingung oder die Variable den Wert 'true' zugeordnet bekommt, wird die Wiedergabe der Audio-Datei begonnen.

9.4.13 Video-Icon (video)

Neben digitalen Filmsequenzen können auch Video-Quellen integriert werden. Abspielgeschwindigkeit, Anfang und Ende lassen sich auch hier frei bestimmen.

9.4.14 Start- und Stopflagge

Durch Platzierung einer Start- und einer Stopflagge im Flußdiagramm lassen sich Teile der Präsentation abspielen.

9.4.15 Farbpalette

Neben der Beschriftung von Icons im Flußdiagramm, dient auch die Einfärbung von Icons zur Übersichtlichkeit. Dabei kann der Entwickler zwischen sechs Farben wählen, um die Icons farblich voneinander abzugrenzen.

Wie schon erwähnt, gestaltet sich das Arbeiten mit den Icons sehr einfach und intuitiv. Durch Drag-And-Drop-Verfahren, werden die Icons im Flußdiagramm platziert. Mit einem Doppelklick auf ein Icon im Flußdiagramm lassen sich seine Attribute betrachten und verändern.

Unterhalb der Hauptfensterleiste mit ihren üblichen Pull-Down-Menüs, befindet sich eine weitere Leiste mit mehreren Steuerknöpfen. Dort findet man neben Druckknöpfen für ein neues Dokument, Laden und Speichern, auch immer wieder benötigte Funktionen zur Bearbeitung (ausschneiden, kopieren, einfügen und Funktion rückgängig machen).

Zusätzlich befindet sich hier ein Auswahlfeld für den zu verwendenden Schriftsatz und zwei Schalter, die den Schrifttyp verändern (bold und italic).

Eine weitere Gruppe von Steuerknöpfen helfen bei der Orientierung im Flußdiagramm. Der erste ermöglicht die komfortable Suche nach Wörtern in den Authorware-Dokumenten. Der zweite Knopf dieser Gruppe verbirgt eine Liste, die die Verbindungen zwischen den Icons im Flußdiagramm aufschlüsselt. Darüber hinaus findet man hier einen Knopf, hinter dem sich Informationen über die verwendeten Variablen verbergen.

Um die Präsentation abzuspielen, betätigt man den Abspiel-Knopf. Mit einem Mausklick auf den Knopf rechts neben dem Abspiel-Knopf öffnet sich ein Fenster, das einem über den Ablauf im Flußdiagramm während des Abspielens informiert (trace window). Dieses Fenster erweist sich bei der Fehlersuche im Flußdiagramm als sehr vorteilhaft, da jedes Icon, welches durchlaufen wird, in diesem Fenster reihenfolgegetreu angezeigt wird. Ein Schalter zur Aktivierung der Hilfe-Seiten für Authorware schließt die Leiste ab.

9.5 Variablen und Funktionen

Durch die Verwendung von Icons, die in einem Flußdiagramm miteinander in Beziehung gebracht werden, lassen sich auf komfortable und einfache Weise komplexe Präsentationen erstellen. Doch besonders bezüglich der Informationsaufbereitung in solchen Präsentationen, stößt man bei dieser Entwurfstechnik bald an die Grenzen des Machbaren. Informationen, die sich während des Ablaufs verändern oder erst berechnet werden müssen, bauen auf Variablen und Funktionen auf. Die Notwendigkeit von Variablen und Funktionen kann an einem einfachen Beispiel leicht nachvollzogen werden.

Bei einem interaktiven Lehrkurs soll dem Kursteilnehmer eine Erfolgsstatistik zur Verfügung stehen. Auf Wunsch kann er sich über den prozentualen Anteil der richtig beantworteten Fragen informieren. Die Prozentzahl soll in einem einfachen Balkendiagramm wiedergegeben werden. Der Kursentwickler kann den Erfolgsbalken nicht einfach mit einem Darstellungs-Icon in das Flußdiagramm einbinden, da ihm die notwendige Information über die Länge des Balken fehlt.

Authorware stellt zur Erfolgsermittlung eine Reihe von Variablen bereit, darunter auch die Variable *PercentCorrect*. Über sie wird der prozentuale Anteil aller Fragen, die richtig beantwortet wurden, ermittelt. Mit dieser Information läßt sich die Länge des Balken bestimmen. Um den Balken mit der korrekten Länge darstellen zu können, wird eine Graphikfunktion benötigt, welche ein Rechteck zeichnet. Die Funktion *Box*(Stiftgröße, *x1*, *y1*, *x2*, *y2*) erfüllt die gewünschte Anforderung. Durch die Verknüpfung der Variable *PercentCorrect* mit der Funktion *Box*, ist der Befehl für den Balken zur Anzeige des aktuellen Prozentanteils komplett:

Box(1, 100, 0, 140, *PercentCorrect*)

Die in *Box* verwendete Variable *PercentCorrect* wird auch Systemvariable genannt. Das bedeutet, daß sie automatisch aktualisiert wird. Authorware stellt dem Entwickler über 200 Systemvariablen zur Verfügung. Diese lassen sich in neun Kategorien gruppieren.

1. Entscheidungsvariablen (Bsp.: *TimesSelected* – Gibt Auskunft, wie oft der letzte Entscheidungspfad eingeschlagen wurde)
2. Dateivariablen (Bsp.: *FileSize* – Größe der aktuellen Datei in Bytes)
3. Rahmenvariablen (Bsp.: *PageCount* – Anzahl der Icons, die mit einem bestimmten Rahmen-Icon verbunden sind)
4. Allgemeine Variablen (Bsp.: *ClickX/ClickY* – Zeigerposition nach dem letzten Mausklick)
5. Graphikvariablen (Bsp.: *LastX/LastY* – Endkoordinaten der letzten Zeichenfunktion)
6. Iconvariablen (Bsp.: *ExecutingIconID* – Identifikationsnummer des aktuellen Icons)
7. Interaktionsvariablen (Bsp.: *ResponseTime* – Reaktionszeit des Anwenders bei der letzten Interaktion)

8. Zeitvariablen (Bsp.: *Date* – Aktuelles Datum)

9. Videovariablen (Bsp.: *VideoFrame* – Anzahl der Bilder, die abgespielt wurden)

Die Möglichkeit zur Definition von eigenen Variablen ist gegeben. Soll zum Beispiel bei dem Lehrkurs die Anzahl der gelöste Fragen nach der Hälfte und nach allen Fragen verglichen werden, so muß die Prozentzahl nach der Hälfte der Fragen gesichert werden. Dies läßt sich durch die Definition einer neuen Variable über folgende Zuweisung bewerkstelligen:

richtig_hälfte:=PercentCorrect

Neu definierte Variablen sind keine Systemvariablen, d.h. der Entwickler muß sich um die Aktualisierung dieser Variablen selber kümmern.

Wie bei den Variablen stellt Authorware auch bei den Funktionen eine ganze Palette dem Entwickler zur Verfügung. Insgesamt werden 13 Kategorien unterschieden:

1. Zeichen-Funktionen (Bsp.: *Code* – Berechnet den ASCII-Wert eines Schriftzeichen.)
2. Dateifunktionen (Bsp.: *ReadExtFile* – Gibt den Inhalt einer Datei zurück.)
3. Rahmenfunktionen (Bsp.: *FindText* – Sucht nach einem Textbereich.)
4. Allgemeine Funktionen (Bsp.: *Initialize* – Setzt alle Variablen auf ihren Initialwert zurück.)
5. Graphikfunktionen (Bsp.: *Box* – Zeichnet ein Rechteck.)
6. Iconfunktionen (Bsp.: *IconType* – Gibt den Typ des ausgewählten Icon zurück.)
7. Sprungfunktionen (Bsp.: *GoTo* – Springt zu einer angegebenen Stelle in der Präsentation.)
8. Programmfunktionen (Bsp.: *if-then* – Ausführung von Instruktionen in Abhängigkeit von einer Bedingung.)
9. Mathematische Funktionen (Bsp.: *SQRT* – Berechnung der Wurzel.)
10. OLE-Funktionen (Bsp.: *OLESetAutoUpdate* – Bestimmt die manuelle oder die automatische Aktualisierung eines OLE-Objekt.)
11. Plattformfunktionen (Bsp.: *CloseWindow* – Schließt das spezifizierte Fenster.)
12. Zeitfunktionen (Bsp.: *DateToNum* – Berechnet aus einem Datum die Anzahl der Tage seit 1.1.1990.)
13. Videofunktionen (Bsp.: *VideoSpeed* – Bestimmt die Abspielgeschwindigkeit.)

Authorware ermöglicht die Verwendung neu definierter Funktionen. In der Windows-Version wird dies durch die Einbindung von DLL-Dateien (dynamic link library) möglich. Beim Macintosh-Pendant wird dies mit HyperCards XCMD oder XCFN bewerkstelligt. Da kein einheitliches Format für beide Plattformen verwendet wird, ist eine Konvertierung von der einen zur anderen Plattform mit Schwierigkeiten behaftet. Authorware fehlt die Fähigkeit DLL in XCMD oder XCFN zu konvertieren und umgekehrt. Es ist dem Entwickler überlassen, die Funktionen für das andere Format neu zu erstellen.

9.6 Kriterienkatalog

Der anschließende Kriterienkatalog soll einen Vergleich mit den anderen, in diesem Seminar besprochenen Autorensystemen, ermöglichen. Die Kritikpunkte wurden in Absprache mit den anderen Seminarteilnehmern erarbeitet und spiegeln die Anforderungen an ein „gutes“ Autorensystem wieder.

9.6.1 Verfügbarkeit

9.6.1.1 Preis

Authorware gehört zur gehobenen Preisklasse der Autorensysteme, obwohl Macromedia den Preis im Vergleich zu älteren Versionen halbierte. Seit der Version 3.5 ist Authorware nur noch als Programmpaket mit dem Namen Authorware Interactive Studio erhältlich und kann nicht mehr einzeln bezogen werden.

Für einen Gesamtpreis von DM 4.659,- erwirbt man zusätzlich zu Authorware den Director 5.0, das Bildbearbeitungsprogramm xRES SE und das Tonbearbeitungswerkzeug SoundForge (SoundEdit 16 plus Deck II bei der Mac-Version). Zusätzlich ist im Windows-Paket der Backstage Designer Plus enthalten, ein Werkzeug zur Erstellung von HTML-Seiten für das Internet. Eine Schulversion ist schon für DM 1.919,- auf dem Markt erhältlich [Gmb96].

9.6.1.2 Hardwareunterstützung

Wie eingangs schon erwähnt, ist das Software-Produkt sowohl für Rechner mit Windows-Betriebssystem (Windows 3.1, Windows95 und NT), als auch für Macintosh-Rechner (68040 und PowerMac) ausgelegt. Auf Rechnern mit 32Bit-Architektur läuft das Programm komplett im erweiterten Modus. Authorware unterstützt die gebräuchlichen Sound- und Graphikkarten. Darüber hinaus werden unterschiedliche MCI-Standards (Overlay, Videodisc, MPEG) auf den Windows-Rechnern unterstützt (Quicktime Overlay bei Macintosh).

Nach dem Director kann nun auch Authorware mit der neuen Version 3.5 Shockwave-Dateien erstellen. Diese lassen sich via Intranet verbreiten und auf einem WWW-Browser mit einem Shockwave-Client abspielen [Mac96a]. Theoretisch sind so die erstellten Präsentationen nicht mehr an Windows- oder Macintosh-Rechner gebunden. Voraussetzung ist nur noch ein entsprechender Shockwave-Client.

9.6.2 Entwicklung

9.6.2.1 Hilfe

Authorware verfügt über eine umfassende Dokumentation. Zum Lieferumfang gehört ein Referenzbuch zum schnellen Nachschlagen (*Authorware Reference*), ein ausführliches Lehrbuch mit zahlreichen Beispielen zum Erlernen der Software (*Using Authorware*), sowie eine über 100 Seiten umfassende Broschüre für die ersten Schritte mit Authorware (*Taking The Plunge*).

Die Dokumentation ist sehr ausführlich und überzeugt durch Verständlichkeit und durch eine schlüssige Gliederung der Themeninhalte. Als großer Nachteil ist zu werten, daß Authorware bisher nur in einer komplett englischen Version verfügbar ist.

Auch die Online-Hilfe ist sehr ausführlich ausgefallen und mit zahlreichen Graphiken bebildert. Eine Suche nach Schlagworten ist möglich. Neben den inzwischen schon üblichen HyperText-sensitiven Hilfeseiten können zusätzlich durch Mausclick Informationen zu Schaltern und Icons aufgerufen werden.

9.6.2.2 Konfiguration der Arbeitsumgebung

Eine Anpassung der Arbeitsoberfläche durch Neugestaltung der Menü-Leisten oder durch Eigendefinition von Knöpfen ist nicht gegeben.

9.6.2.3 Format-Unterstützung

Authorware unterstützt auf Windows-Plattformen folgende Datei-Formate [Inf96]:

- Text
 - ASCII, Rich Text Format (.RTF)
- Bild

Windows Bitmap (.BMP), Device Independent Bitmap (.DIB), Encapsulated PostScript Files (.EPS), Macintosh PICT (.PCT), ZSoft Paintbrush (.PCX), Windows RLE (.RLE), Tagged Image File Format (.TIFF), Windows Metafile (.WMF)

- Audio
AIFF (.AIF), Macintosh PCM (.PCM), WAVE (.WAV)
- Video
Video for Windows (.AVI), QuickTime for Windows (.MOV), MPEG Movies (.MPG)
- Animation
Bildfolge in Windows Bitmap oder Device Independent Bitmap (.BMP, .DIB), Autodesk Animator File und Autodesk Animator Pro File (.CEL, .FLC, .FLI), Director 4.0 Movies (.DIR, .DXR)

Die Macintosh-Version unterstützt die folgenden Formate:

- Text
ASCII, Rich Text Format
- Bild
Macintosh PICT
- Audio
AIFF, Macintosh PCM, WAVE
- Video
Quicktime for Macintosh
- Animation
Director 4.0 Movies, PICS Movies

9.6.2.4 Unterstützung von Teamarbeit

Das Autorensystem wurde nicht konkret für die Arbeit in einer Gruppe ausgelegt. Es fehlen eine Versionsverwaltung sowie Protokollmechanismen für Gemeinschaftsprojekte.

9.6.2.5 Wiederverwendbarkeit

Das Verwalten von Medienobjekten, Programmroutinen oder auch Präsentationssequenzen in Bibliotheken ist möglich und vereinfacht so die Wiederverwendbarkeit von Entwurfsfragmenten für weitere Entwicklungen.

9.6.2.6 Skripten

Eine Verwendung von Programmroutinen ist möglich und bei komplexeren Präsentationen sogar notwendig. Die Einbindung erfolgt über externe Dateien (DLL unter Windows, XCMD bzw. XFCN auf Macintosh).

9.6.2.7 Wartung

Die Wartung der Präsentationen unter Authorware gestaltet sich problematisch. Durch die Darstellung des Präsentationsablauf in einem Flußdiagramm mit Icons und unterschiedlichen Ebenen kann das Wiederfinden einer bestimmten Stelle sehr zeitraubend verlaufen. Die Entwickler von Authorware versuchten diesen Nachteil mit einer Suchfunktion und einem sogenannte Trace-Window wieder wettzumachen. Trotz dieser sehr hilfreichen Funktionen bedarf das Programm hier einer Nachbesserung, um die Wartung noch komfortabler zu gestalten.

9.6.2.8 Übersetzung

Eine Aussage bezüglich des Zeitbedarfs für die Erstellung der Präsentationsmodule und der Effizienz kann nicht gemacht werden, da diese Funktion in der untersuchten Demoversion nicht unterstützt wird.

9.7 Fazit

Die Besprechung aller Eigenschaften und Funktionalitäten von Authorware würden den Rahmen dieser Seminararbeit sprengen. Auf viele Details, wie z.B. Datenbank-Unterstützung, Überblendungstechniken, Graphik-Bibliotheken usw., mußte verzichtet werden. Wichtiger war es, einen allgemeinen Überblick über dieses mächtige und reife Autorensystem zu vermitteln, das trotz einiger Schwachpunkte zu den besten zählt.

9.8 Weiterführende Informationsquellen

Die folgende Liste von Internet-Adressen soll dem interessierten Leser bei der Vertiefung in die Thematik zur Seite stehen.

- Macromedia Informations-Server <http://www.macromedia.com/>
- Macromedia User Journal <http://www.hyperstand.com/MUJ/muj.html>
- Macromedia Authorware Seiten <http://www.macromedia.com/software/ais>
- Authorware Online Manual <http://info.psy.uva.nl/COO/AOM/AOMhome.html>
- The AWARE Page <http://www.hvu.nl/aware>
- Authorware Newsgroup LISTSERV@CC1.KULEUVEN.AC.BE
- FTP-Server für Authorware
<ftp://pa.cc.kuleuven.ac.be/pub/authorware/>
<ftp://ftp.ccc.nottingham.ac.uk/pub/dlls/>
- Allgemeine Infos zu Multimedia
<http://viswiz.gmd.de/MultimediaInfo>
<http://www.hightext.de/aktuell.htm>

Eine CD-ROM mit einer Authorware-Testversion und weiteren Demo-Programmen kann von der deutschen Vertriebsstelle für Macromedia-Produkte bezogen werden.

Anschrift: MACROMEDIA
 c/o PMA GmbH
 Postfach 701365
 22041 Hamburg

9.9 Literatur

- [DPR96] Thomas J. Schult Dr. Peter Reimann. *Virtuell lehren leicht gemacht?* c't, Seiten 270-278, Januar 1996.
- [Gmb96] Softline GmbH. *Softline Katalog 12*. Software-Produktkatalog, Softline GmbH, Oktober 1996.
- [Inf96] Macromedia Informationsserver. *Authorware - overview & key features*.
 URL: <http://www.macromedia.com/software/ais/>, 5.11.1996.
- [Mac96a] Inc. Macromedia. *Shockwave for Authorware*.
 URL: <http://www.ttm.fi/tuotteet/Macromed/Shockwave/authorinfo.htm>, 16.4.1996.

-
- [Mac96b] Inc. Macromedia. *Authorware 3.5*. Produktbroschüre, Macromedia, Inc., 1996.
- [mul95a] multiMEDIA. *Verdopplung bei deutschen CD-ROM-Titel*.
URL: <http://www.hightext.de/archiv/1295.htm>, Dezember 1995.
- [mul95b] multiMEDIA. *Deutscher Edutainment-Markt wächst auf 1,3 Milliarden Mark*.
URL: <http://www.hightext.de/archiv/1095.htm>, Oktober 1995.
- [mul96] multiMEDIA. *Laufwerk- vor CD-Wachstum*.
URL: <http://www.hightext.de/archiv/0296.htm>, Januar 1996.

10.1 Übersicht

Die CD ist ein beliebtes Speichermedium – sei es für Audio, Video oder andere Daten. Der Grund dafür liegt auf der Hand. Sie ist kostengünstig herzustellen, kann viele Daten speichern und die Daten „halten länger“ als bei magnetischen Speichern wie der Diskette oder der Festplatte.

Dieser Artikel gibt einen kleinen Überblick darüber, wie Informationen auf einer CD gespeichert und wieder ausgelesen werden. Außerdem wird erläutert, wie der Herstellungsprozeß einer CD abläuft. Danach beschreibe ich CD Standards und zeige, wie sie aufeinander aufbauen. Auch die bei CDs verwendeten Fehlerkorrekturverfahren stelle ich hier vor.

Da die CD für multimediale Anwendungen nicht genügend Speicherplatz bietet, wird sie von der DVD (Digital Versatile Disc) bald verdrängt werden. Die Besonderheiten der DVD sind hier aufgezeigt. Danach wird ein neuer Ansatz zur 3-dimensionalen Speicherung von Daten vorgestellt.

Da aus ökologischen Gründen eine CD nicht einfach weggeworfen werden soll, widmet sich dieser Artikel noch dem CD-Recycling.

10.2 CD-ROM Hardware

10.2.1 Aufbau einer CD

Ein durchsichtiger Kunststoff macht den größten Teil der Gesamtdicke von 1.2 mm einer CD-ROM aus (vgl. Abbildung 10.1). Die Daten sind physikalisch in Pits (Vertiefungen) enthalten, die in den Kunststoff geprägt werden. Darauf ist eine Metallschicht aus Aluminium, Silber oder Gold aufgedampft. Zum Schutz dieser metallisierten Oberfläche wird eine Schutzschicht (Lackschicht) aufgebracht, auf die das Label der CD gedruckt wird. Von der Unterseite gesehen wirken die Löcher wie kleine Erhebungen, weshalb oft auch von Bergen statt von Löchern bzw. Vertiefungen gesprochen wird. Die Pits sind so winzig, daß sie weder mit bloßen Auge noch mit einer Lupe zu erkennen sind, sondern nur mit einem

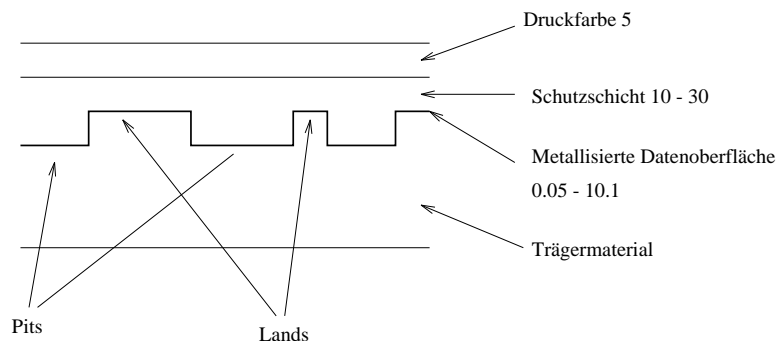


Abbildung 10.1: Aufbau einer CD (Angaben in μm)

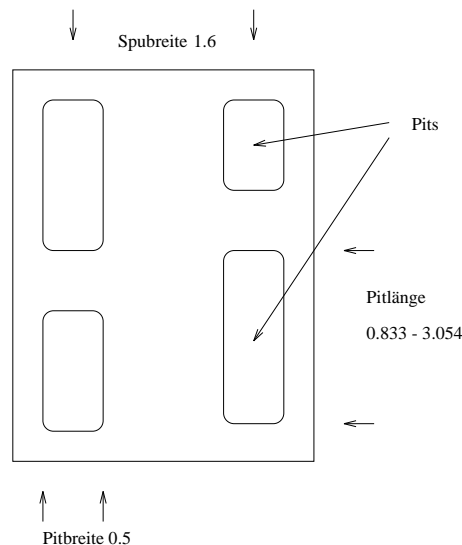
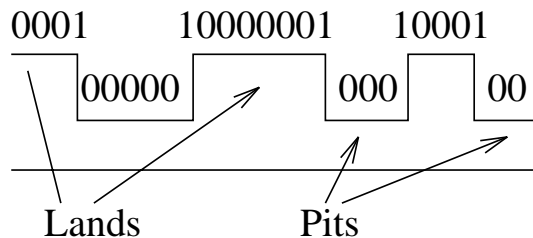
Abbildung 10.2: Maße der Pits (Angaben in μm)

Abbildung 10.3: Kodierung mit Pits und Lands

Rasterelektronenmikroskop. Die genauen Maße sind Abbildung 10.2 zu entnehmen. [She89]

Eine CD enthält eine fortlaufende Spur von Pits, die sich spiralförmig von innen nach außen windet. Das hat den Vorteil, daß problemlos auch kleinere CDs abgespielt werden können und Beschädigungen am äußeren Rand die Daten nicht unbrauchbar machen, weil ein kleiner Sicherheitsabstand eingehalten wird. Jede Pitkante, gleichgültig ob am Anfang oder am Ende, steht für eine 1; sämtliche Flächen dazwischen, innerhalb oder außerhalb eines Pits, entsprechen einer 0 (vgl. Abbildung 10.3). [Nad94]

Abgetastet werden die Daten mit einem Laser, der unterhalb der CD angebracht ist und dessen Strahl von der Metallschicht reflektiert wird (vgl. Abbildung 10.4). Da die Pithöhe $\frac{\lambda}{4}$ beträgt, erfährt der Laserstrahl eine Phasenverschiebung um $\frac{\lambda}{2}$. Daher kommt es zwischen dem von einem Pit reflektierten Strahl und einem von der Oberfläche reflektierten Strahl zu Interferenzen und gegenseitiger Auslöschung. Die Datendichte der Pitspur ist am Anfang der Spirale die gleiche wie am Ende. Daher muß die CD mit konstanter Lineargeschwindigkeit von 1.2 - 1.4 Meter pro Sekunde rotieren und die Geschwindigkeit der CD von innen nach außen von 500 auf 200 Umdrehungen pro Minute abnehmen.

Die kleinste Dateneinheit auf einer CD wird ein Frame genannt. Jeder Frame enthält Nutzdaten, die zugehörigen Paritätsdaten, ein Synchronisationswort sowie einen Subcode von 8 Bits und besteht aus insgesamt 291 Bits (vgl. Tabelle 10.1). Aufgrund der Kodierung (vgl. Tabelle 10.2) müssen hierfür 588 Bits gespeichert werden. 98 Frames bilden einen Sektor, der die kleinste adressierbare Einheit ist. [She89]

Sync	Subcode	User Data	Parity	User Data	Parity
27	8	96	32	96	32

Tabelle 10.1: Frame (Angaben in Bits)

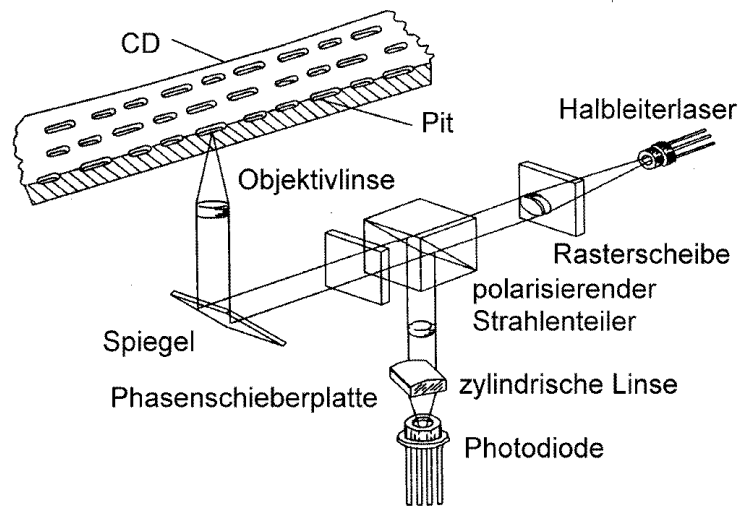


Abbildung 10.4: Abtasteinheit

10.2.2 EFM-Modulation

Um ein einwandfreies Auslesen einer CD zu gewährleisten, müssen zwischen zwei Einsen mindestens 2 Nullen, aber maximal 11 Nullen, stehen. Da eine gewöhnliche 8-Bit Kodierung diese Anforderungen nicht erfüllt, werden die 8-Bit Muster auf entsprechend ausgewählte 14-Bit Muster abgebildet. Diesen Vorgang nennt man EFM-Modulation (Eight-to-Fourteen-Modulation). Ein paar Zuordnungen sind in Tabelle 10.2 aufgeführt. Das ist auch der Grund, weshalb die Länge verschiedener Pits zwischen 0.833 und $3.054 \mu\text{m}$ variiert. Jetzt kann es noch vorkommen, daß zwei aufeinanderfolgende 14-Bit Muster die oben erwähnte Einschränkung nicht erfüllen. Daher werden zwei 14-Bit Muster mit 3 Koppelbits verbunden. Zwei Koppelbits (immer Nullen) sind notwendig, um eine mögliche Aufeinanderfolge von Einsen in den seriellen Blöcken zu verhindern. Das zusätzliche Koppelbit (entweder 1 oder 0, abhängig

8-Bit Muster	14-Bit Muster (EFM)
00000000	01001000100000
00000001	10000100000000
00000010	10010000100000
00000011	10001000100000
00000100	01000100000000
00000101	00000100010000
00000110	00010000100000
00000111	00100100000000
00001000	01001001000000
00001001	10000001000000
00001010	10010001000000

Tabelle 10.2: Beispiele zur EFM-Modulation

vom vorangehenden und nachfolgenden Muster) hat die Aufgabe, die Taktsynchronisation zu erleichtern und tieffrequente Signalkomponenten zu unterdrücken. [Poh94]

10.2.3 Pressen und Brennen einer CD

Die wesentlichen Fertigungsstufen umfassen Mastering, Galvanisieren, Pressen, Metallisieren, Lackieren, Stanzen und Drucken. Beim Mastering wird eine kreisrunde, vollkommen ebene Glasscheibe (Glas-Vater) mit einer lichtempfindlichen Beschichtung von einem Laserstrahl mit dem Muster der Pits belichtet. Nach dem Wegätzen der belichteten Teile wird die Scheibe mit Silber beschichtet (Metall-Vater). In einem Nickelbad wird dann galvanisch ein Negativ (Metall-Mutter) hergestellt, von dem mehrere Matrizen angefertigt werden, die als Preßform beim automatischen Pressen der Compact Disc dienen. Die Seite mit dem Pitmuster wird nach dem Pressen im Vakuum mit Aluminium bedampft, um eine reflektierende Schicht zu erhalten. Danach wird noch eine Schutzschicht aufgebracht. Sämtliche Bearbeitungsvorgänge erfordern hohe Präzision und sind nur bei Reinluftbedingungen durchführbar. Die korrekte Position des einzustanzenden Mittel Lochs wird den Spuren der Compact Disc entsprechend optoelektronisch ermittelt. Eine andere Möglichkeit, Daten auf eine CD zu bekommen, ist die mit Hilfe eines CD-Writers. Die dazu notwendigen CD-Rohlinge werden CD-R (CD-Recordable) genannt. Die CD-R ist fast wie eine normale, gepreßte CD aufgebaut. Sie besteht aus einer Polycarbonat-Scheibe, einer Reflexionsschicht und einem Schutzlack auf der Labelseite. Außerdem befindet sich bei der CD-R zwischen Polycarbonat und Reflexionsschicht noch ein Farbstoff. Dieser Farbstoff wird beim Schreiben der Informationen verändert. Und zwar wird dann die relativ transparente Farbschicht punktuell kurzzeitig erhitzt und eine chemische Reaktion ausgelöst, die die Lichtdurchlässigkeit des Farbstoffes verändert und eine Art Bläschen erzeugt. Dabei wird auch das Reflexionsmaterial an der Stelle leicht angehoben und das Polycarbonat „angeschmort“. Die so durch den Laser bewirkten Markierungen sind wie Pits auf einer gepreßten CD unterschiedlich lang und haben die gleiche Wirkung: Sie verursachen eine Phasenverschiebung und führen so zu Interferenzen.

Der CD Rohling enthält bereits den sogenannten Pregroove. Diese durchgehende, wellenförmige Leerspur wurde mit einer Frequenz von 22.05 kHz aufmoduliert. Sie ist für die Spurführung beim Schreiben nötig; aus der Frequenz gewinnt der Schreiber Informationen über die Umdrehungsgeschwindigkeit und regelt sie so, daß die CD mit konstanter Datendichte beschrieben wird. Abbildung 10.5 zeigt eine CD-R mit Pregroove und eine gepresste CD. [Poh94]

10.3 CD-ROM Standards und Formate

Ein Format gibt Aufschluß darüber aus wievielen Bytes ein Sektor zusammengestzt ist und welche Daten (Audio, Video, ...) in einem Sektor stehen. Definiert sind die aufeinander aufbauenden Formate in den sogenannten *Rainbow Books* (vgl. Abbildung 10.6). [Nad94]

10.3.1 CD-DA

Der erste Standard für eine CD ist der einer Audio-CD. Dieser wird als CD-DA (Digital Audio) bezeichnet und ist im *Red Book* beschrieben. Er ermöglicht die Speicherung von bis zu 74 Minuten unkomprimierten digitalen Audio-Daten mit einer festen Frequenz von 44.1 kHz Sampling bei 16 Bit und Stereokanal. Eine Audio-CD besteht aus mehreren Audiotracks, wobei meistens ein Track einem Song entspricht. Jeder Track ist außerdem noch in Sektoren unterteilt, die eine Länge von 2352 Bytes bzw. $\frac{1}{75}$ Sekunde haben. Adressiert wird ein Sektor mit Minute:Sekunde:Sektor.

Da die Wahrscheinlichkeit sehr groß ist, daß z. B. durch Kratzer oder Schmutz eine CD nicht mehr korrekt gelesen werden kann, ist im *Red Book* eine Fehlerkorrektur enthalten. Diese Fehlerkorrektur benutzt CIRC (Cross-Interleaved-Reed-Solomon-Code). Die Daten werden in Blöcke zusammengefaßt und von jeder Zeile und Spalte eine Checksumme gebildet. Tritt ein Fehler auf, so kann anhand der falschen horizontalen und vertikalen Checksumme der Fehler lokalisiert und korrigiert werden (vgl. Tabelle 10.3). Flächenfehler, die mit CIRC nicht mehr korrigiert werden können, können auf unterschiedliche Weise kompensiert werden (vgl. Tabelle 10.4):

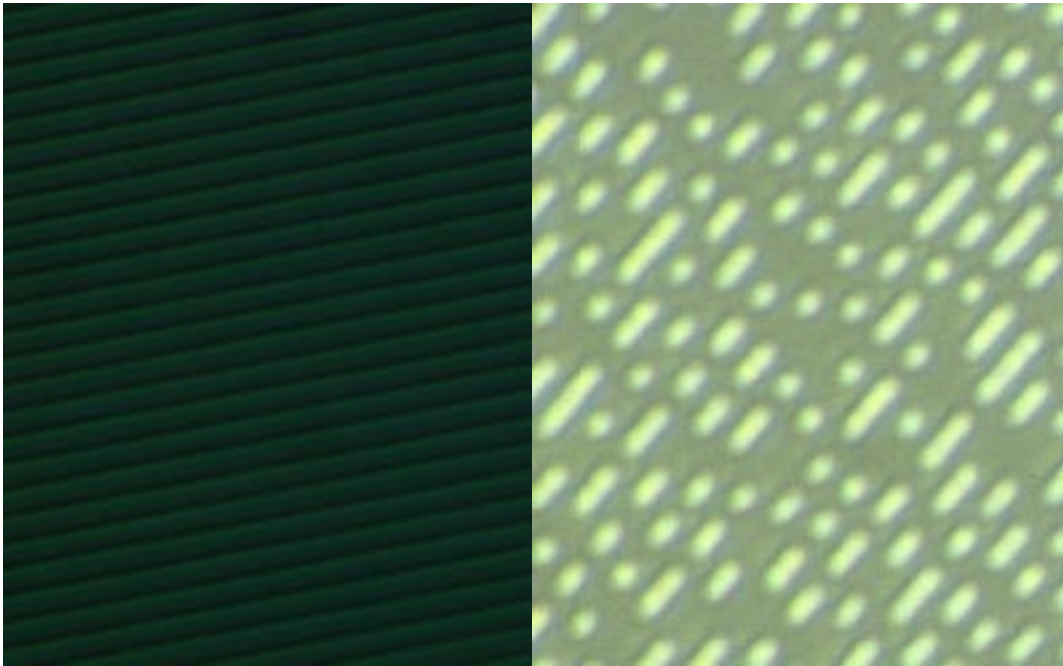


Abbildung 10.5: Pregroove einer CD-R und eine gepresste CD

1	2	3	6	1	2	3	6
4	5	6	15	4	4	6	14
7	8	9	24	7	8	9	24
12	15	18		12	14	18	
ohne Fehler				mit Fehler			

Tabelle 10.3: Beispiel für Blockchecksummen

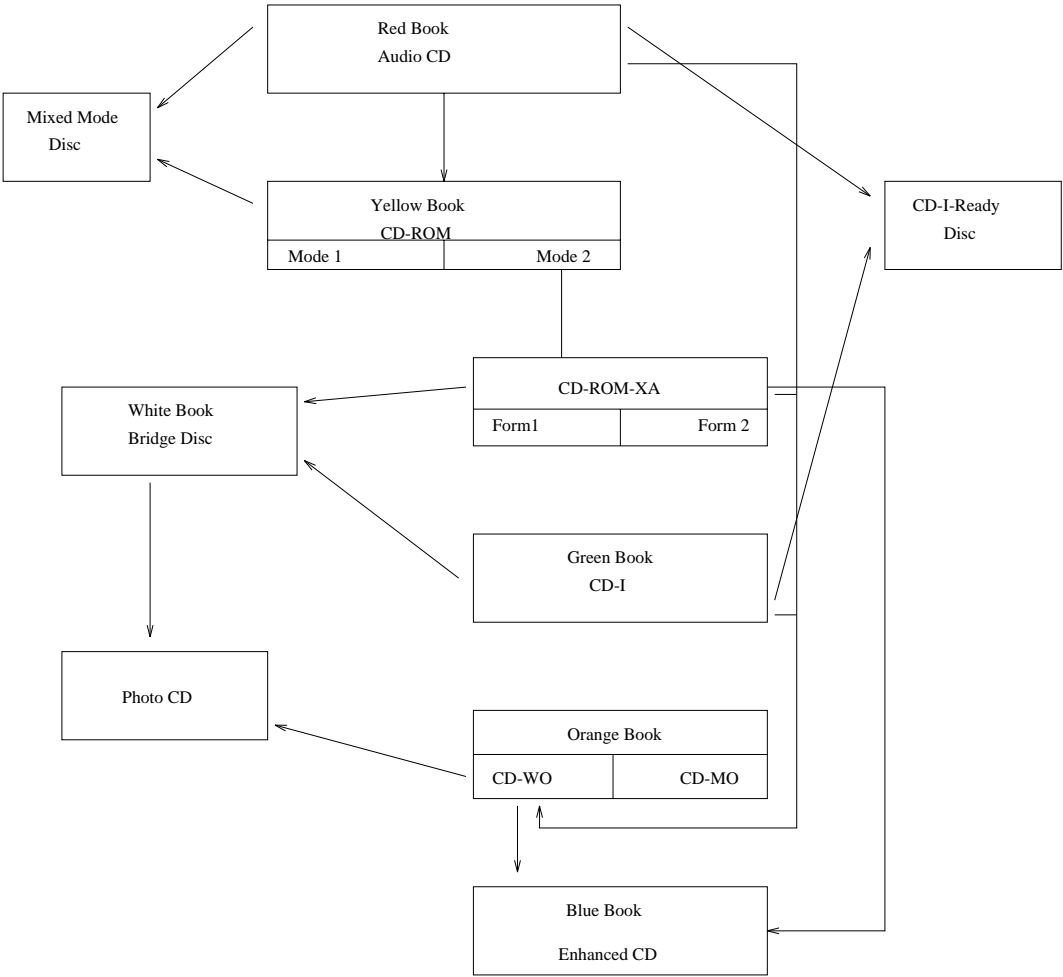


Abbildung 10.6: CD Formate

Max. vollständig korrigierbare Fehlerlänge	4000 Datenbits entspricht 2.27 mm Spurlänge
Max. interpolierbare Länge im schlimmsten Fall	13700 Datenbits entspricht 8.5 mm Spurlänge

Tabelle 10.4: Daten zu CIRC

Sync	Header	User Data	EDC	Blanks	ECC
12	4	2048	4	8	276

Tabelle 10.5: Sektor im CD-ROM Mode 1

- den Wert interpolieren
- den vorherigen Wert halten
- während der fehlerhaften Stelle den Lautsprecher stummschalten

10.3.2 CD-ROM

Das CD-ROM Format ist im *Yellow Book* definiert und unterscheidet zwei Modi: Mode 1 (vgl. Tabelle 10.5), die gängige Daten-CD, und Mode 2 (vgl. Tabelle 10.6) mit verringerter Fehlerkorrektur und erhöhter Kapazität. Die ersten 16 Bytes werden benutzt, um einen Sektor zu adressieren; danach folgen die Nutzdaten. Außerdem verwendet Modus 1 288 Bytes eines Sektors für eine zusätzliche Fehlererkennungs- und Korrektorebene (EDC/ECC). Dieser Zusatz zu dem auf allen CDs enthaltenen Korrekturcode (CIRC) ermöglicht eine Datensicherheit, die bei sich plötzlich ändernden Daten, anders als dies bei Musik der Fall wäre, notwendig wird. Tritt bei einer Audio-CD ein Fehler auf, der nicht korrigiert werden kann, so wird der Wert interpoliert oder der Ausgang stummgeschaltet. Bei einer CD-ROM mit Computerdaten wäre das verheerend. Ein guter Fehlerkorrekturcode benutzt neben Parität auch noch andere Techniken wie Interleaving, um möglichst alle Fehler beim ersten Lesen zu korrigieren. Interleaving ist eine Art Verschachtelungsmethode und wird als Schutz gegen das sehr wahrscheinliche Auftreten von Flächenfehlern benutzt. Datenwörter des Bitstroms werden vor der Aufzeichnung so umverteilt, daß aufeinanderfolgende Worte auf dem Speichermedium niemals benachbart sind. Ein auftretender Fehler (z. B. durch ein Staubkorn verursacht) verhindert korrektes Lesen einer Reihe von aufeinanderfolgenden Blöcken. Wenn jedoch beim De-Interleaving die vermischten Wörter wieder in ihre ursprüngliche Reihenfolge gesetzt werden, ist der fehlerhafte Teil in einzelne Abschnitte isoliert und kann so viel leichter korrigiert werden. [She89]

10.3.3 Mixed-Mode-CD

Eine Mixed-Mode-CD ist ein Format, das sowohl CD-DA Audiotracks als auch CD-ROM Datentracks enthält. Der erste Track wird immer als CD-ROM Mode 1 Track definiert und die nachfolgenden als CD-DA Audiotracks.

Sync	Header	User Data
12	4	2336

Tabelle 10.6: Sektor im CD-ROM Mode 2

Sync	Header	Subheader	User Data	EDC	ECC
12	4	8	2048	4	276

Tabelle 10.7: Sektor im CD-ROM Mode 2, XA Form 1

Sync	Header	Subheader	User Data	EDC
12	4	8	2324	4

Tabelle 10.8: Sektor im CD-ROM Mode 2, XA Form 2

10.3.4 CD-ROM-XA

Der CD-ROM Standard ist später um das sogenannte CD-ROM-XA Format (Extended Architecture) erweitert worden. Das CD Format erlaubt die Speicherung sowohl von Computerdaten und -programmen (Form 1, vgl. Tabelle 10.7) als auch von Audio- und Videodaten (Form 2, vgl. Tabelle 10.8) innerhalb eines Tracks. Erreicht wird das durch eine Verschachtelung der Sektoren, wobei zur Unterscheidung der Datentypen (Form 1/2) der Subheader verwendet wird.

10.3.5 CD-I

Die CD-I (CD-Interactive) ist gemäß dem *Green Book* standardisiert. Auf ihr können sämtliche Datentypen (Audio, Video, usw.) gespeichert werden. Der Unterschied zur CD-ROM ist, daß die CD-I in Echtzeit abgespielt wird. D. h. Musik, Video, Text, usw. müssen synchronisiert sein. Wie bei CD-ROM-XA sind die Datentypen auf der CD verschachtelt gespeichert und werden mit Hilfe des Subheaders identifiziert. Das Sektorformat ist gleich dem des CD-ROM-XA Standards. CD-I Discs können nur mit einem CD-I Player wiedergegeben werden. [Poh94]

10.3.6 CD-I-Ready

Das CD-I-Ready Format ermöglicht die Speicherung von CD-I und CD-DA Tracks auf einer CD und die Wiedergabe auf beiden Arten von Playern (CD-I und CD-DA Player). Jedoch kann der Audio-CD Player nur die Audio Daten wiedergeben, der CD-I Player dagegen beide Formate. Audio-CDs haben einen 2 bis 3 Sekunden langen Pregap am Anfang des ersten Tracks. Im Pregap stehen Informationen, die zum Abspielen von CD-I Daten notwendig sind. Ein Audio-CD Player überspringt den Pregap und findet daher nur die Audio-Daten.

10.3.7 CD-I-Bridge und Video-CD

Das *White Book* definiert wie CD-I Daten auf CD-ROM-XA CDs geschrieben werden und ermöglicht so auch die Wiedergabe von beschreibbaren CDs wie der Photo-CD auf CD-ROM-XA und CD-I Systemen. Als kleine Einschränkung können die Daten nur in CD-ROM-XA Mode 2 oder in CD-DA geschrieben werden.

Das *White Book* dient auch als Grundlage für die Video-CD. Mit MPEG komprimiert können Filme mit einer Dauer von 74 Minuten auf einer solchen CD gespeichert werden.

10.3.8 CD-MO und CD-WO bzw. CD-R

Der *Orange Book* Standard Teil I beschreibt zwei auf einer CD-MO (CD-Magneto-Optical) zu findende Gebiete. Erstens kann eine CD-MO einen Bereich enthalten, der nur gelesen werden kann und zweitens einen Bereich, der wiederbeschreibbar ist. Der wiederbeschreibbare Teil kann von einem herkömmlichen CD-Player nicht gelesen werden, weil die Daten mit einem anderen Verfahren aufgezeichnet werden.

Lead-in and TOC	Information	Lead-out	Lead-in and TOC	Information	Lead-out	...
	Session 1		Session 2			

Tabelle 10.9: Multisession CD

Base/16	192 x 128 Punkte
Base/4	384 x 256 Punkte
Base	768 x 512 Punkte
Base 4	1536 x 1024 Punkte
Base 16	3072 x 2048 Punkte

Tabelle 10.10: Auflösungen einer Photo-CD

Teil II des *Orange Book* beschreibt die CD-WO (CD-Write-Once) bzw. CD-R (CD-Recordable). Der Standard erlaubt das Beschreiben einer CD-R in mehreren Sessions. Eine in mehreren Sessions beschriebenen CD nennt man Multisession CD (vgl. Tabelle 10.9).

10.3.9 Photo-CD

Die Photo-CD ist eine multisessionfähige Bridge Disc, die für die Speicherung von Bildern, Photos und Dias usw. eingesetzt wird. Zum Speichern der Bilder stehen mehrere Auflösungen zur Verfügung (vgl. Tabelle 10.10). Die Photo-CDs sind über einen CD-I Player oder über ein XA-fähiges CD-ROM Laufwerk abspielbar. [Nad94]

10.3.10 Enhanced-Music-CD

Die Enhanced-Music-CD ist im *Blue Book* definiert und ersetzt die Mixed-Mode-CD. Die Audiotracks liegen jetzt vor den Datentracks und nicht, wie bei der Mixed-Mode-CD, dahinter. Das hat den Vorteil, daß ein Audio-CD-Player nicht irrtümlich auf die Datentracks zugreift und dadurch Lautsprecher oder Verstärker beschädigt werden können. Die Enhanced-Music-CD ist eine Multisession-CD mit zwei Sessions. Die erste ist entsprechend einer CD-DA und die zweite entsprechend einer CD-ROM-XA nach den Standards CD-ROM, CD-WO Teil 2 spezifiziert. Das Dateisystem der 2. Session entspricht ISO 9660 und folgt dem CD-ROM-XA Standard. [Bör96]

10.3.11 Filesysteme

10.3.11.1 ISO 9660

Das ISO 9660 Dateisystem löste das High-Sierra Dateisystem ab. Sektoren werden zu logischen Records zusammengefaßt und dann diesen Records Dateien zugewiesen. ISO 9660 benutzt ähnlich MS-DOS ein hierarchisch aufgebautes Inhaltsverzeichnis in Baumstruktur. Die Tiefe des Baumes darf maximal 8 betragen. Damit viele Firmen mit diesem Dateisystem einverstanden waren, einigte man sich auf den gemeinsamen Nenner, damals war das MS-DOS. Daher sind die Dateinamen auf 8+3 Zeichen beschränkt. Das ISO 9660 Dateisystem ermöglicht es, jedem Betriebssystem CD-ROMs zu lesen.

10.3.11.2 Rock-Ridge-Extension

Die Rock-Ridge-Extension erweitert das ISO 9660 um lange Dateinamen und Attribute.

10.3.11.3 Apple HFS

Darüber hinaus kann fast jedes Betriebssystem ein eigenes CD-Filesystem lesen, das dem der eigenen Festplattenpartition im wesentlichen entspricht, z. B. Apple HFS.

10.3.12 Hybrid-CD

CDs, die sowohl ISO 9660- als auch HFS-Dateisysteme enthalten, nennt man Hybrid-CDs. Es ist ein Bridge-Disc Format zwischen Mac HFS und ISO 9660. Beide Partitionen befinden sich gleichzeitig auf einer CD. Jedes Betriebssystem greift auf die für ihn vorgeschriebene Partition zu. Der Mac kann auch auf gemeinsame Daten der ISO 9660 Partition zugreifen.

10.4 DVD

Immer speicherhungrigere, multimediale Anwendungen verlangen nach noch mehr Kapazität auf den Datenträgern. Auf eine Video-CD passen nicht einmal ganze Spielfilme, sondern der Spielfilm muß auf mehrere CDs verteilt werden. Ein Standard wurde geschaffen und als DVD (Digital Versatile Disc) bezeichnet. Dieser neue Standard ist so vielversprechend, daß er womöglich Videokassette, Laserdisc, CD-ROM und Audio-CD ersetzen könnte. Die DVD hat die gleichen Maße wie eine CD. Sie hat einen Durchmesser von 120 mm und eine Dicke von 1.2 mm. Die DVD ist jedoch sehr viel dichter beschrieben als eine herkömmliche CD. Hinzu kommt, daß die digitalen Informationen in zwei Lagen übereinander liegen können (vgl. Tabelle 10.11) [Doy96]. Wenn nötig, wird zusätzlich neben der Vorder- auch die Rückseite des Datenträgers genutzt. So passen bis zu 26-mal mehr Daten auf die DVD als auf eine herkömmliche CD:

- 4.7 GB bei single-sided, single-layer disc
- 8.5 GB bei single-sided, double-layer disc
- 9.4 GB bei double-sided, single-layer disc
- 17 GB bei double-sided, double-layer disc

Bereits die single-sided, single-layer disc bietet die Möglichkeit, Spielfilme mit 133 Minuten Spieldauer zu speichern. [Son96]

Die geplanten DVD-Formate lösen wahrscheinlich die entsprechenden CD-Formate ab. Folgende DVD-Formate wird es geben:

DVD-Video Speicher für Bild- und Tonwiedergabe sowie für interaktive Spiele. Eine Seite des Mediums faßt Filme mit Längen von bis zu 133 Minuten.

DVD-Audio Ersatz für die Audio-CD mit deutlich besserer Klangqualität und einer maximalen Spielzeit von 7.5 Stunden.

DVD-ROM CD-ROM Nachfolger mit Speicherkapazität zwischen 4.7 und 17 GB.

DVD-R Einmal beschreibbarer Datenträger mit Kapazitäten zwischen 4 und 7.8 GB.

DVD-RAM Für 1998 geplantes, mehrfach beschreibbares Medium.

Zur Zeit gibt es noch Probleme bei den Herstellern der DVD-Player, weil die Filmindustrie einen Ländercode und ein Verschlüsselungsverfahren zum Kopierschutz fordert. Der Ländercode soll dazu dienen, daß die Verwertungskette der Filmindustrie nicht zerstört wird. Ein DVD-Player, der beispielsweise in Deutschland verkauft wird, soll dann eine für den US-Markt hergestellte DVD-Video nicht wiedergeben können. [Wil96]

10.5 Neuer Ansatz

Auf herkömmlichen CDs wird zur Speicherung nur die Oberfläche verwendet. Ein neues Forschungsprojekt zeigt nun, daß man die Daten auch dreidimensional speichern kann. Um dies zu erreichen, werden synthetische Farbstoffe benutzt, die ihre Fluoreszenz ändern können. Die beiden digitalen Zustände werden also durch eine vorhandene oder nicht vorhandene Fluoreszenz charakterisiert. Es können etwa

	CD	DVD
Durchmesser	120 mm	120 mm
Dicke	1.2 mm	1.2 mm
Spurabstand	1.6 μm	0.74 μm
min. Pit-Größe	0.83 μm	0.4 μm
CLV	1.2 m/s	4.0 m/s
Daten-Layers	1	1 oder 2
Kapazität	etwa 680 MB	bis 17 GB

Tabelle 10.11: Technische Daten einer DVD

ein Terabyte an Daten pro Kubikzentimeter gespeichert werden. Kleine Testläufe wurden schon durchgeführt. Und zwar wurden wenige Sekunden lange Videosequenzen eines Comic-Streifens abgespielt. [Nöl96]

10.6 Recycling von CDs

Ein nicht vernachlässigbares Problem ist die Entsorgung von ausgedienten CDs, da aufgrund der geringen Herstellungskosten viele CDs hergestellt werden. Auch werden CDs in Umlauf gebracht, die nur einen Bruchteil der zur Verfügung stehenden Kapazität ausnutzen.

Eine CD ist ein Verbundstoff, der aus Polycarbonat, Aluminium und etwas Lack besteht. Hergestellt werden die CD-Rohlinge aus Polycarbonat, anschließend mit Aluminium bedampft und dann mit einer Lackschicht versehen. Der Recycling-Prozeß läuft genau in umgekehrter Reihenfolge ab. So werden durch Ätzen mit Natronlauge Aluminium, Lack und Polycarbonat wieder getrennt. Jedoch enthält das Polycarbonat bei diesem Verfahren noch Aluminium und Lack und ist daher nicht mehr für die Herstellung von neuen CDs geeignet, sondern nur noch für Gehäuse oder ähnliches.

Eine weitere Methode CDs zu recyceln ist, die Scheiben mechanisch mit einem Messer abzuschälen. Da das abfallende Polycarbonat sehr hochwertig ist, kann es direkt für die Produktion neuer CDs verwendet werden.

Ganz gelöst ist das Entsorgungsproblem noch nicht, weil ein paar Reststoffe – eine Masse aus Aluminium, Lackresten und etwas Polycarbonat – übrigbleiben, die auf einer Deponie „entsorgt“ werden müssen. [Hur96]

10.7 Literatur

- [Bör96] Wolfgang Börner. Musik mit Schuß: CD-Extra – die neue interaktive Musik-CD. *c't*, (8):262 ff, August 1996.
- [Doy96] Bob Doyle. DVD Story. <http://www.atvgroup.com/DigVideo/DVD/DVDstory.html>, 1996.
- [Hur96] Frank Hurtz. Silberlinge sind viel mehr als nur ein Abfallprodukt. *COMPUTER ZEITUNG*, (35):17, August 1996.
- [Nad94] Michael Nadeau. *The BYTE Guide to CD-ROM*. Mc Graw-Hill, 1994.
- [Nöl96] Arno Nöldechen. Farbstoffe erhöhen drastisch die Kapazität von CD-ROMs. *COMPUTER ZEITUNG*, (41):34, Oktober 1996.
- [Poh94] K. Pohlmann. *Compact Disc Handbuch*. iwt, 1994.
- [She89] Chris Sherman. *CD-ROM Handbook*. Mc Graw-Hill, 1989.
- [Son96] Sony. Technical Information. <http://www.sel.sony/SEL/consumer/dvd/>, 1996.

- [Wil96] Michael Wilde. Hollywood vermässelt der DVD das Weihnachtsgeschäft. *COMPUTER ZEITUNG*, (48):18, November 1996.



Kapitel 11

Rechtliche Aspekte elektronischer Veröffentlichung

Karsten Kutschera

11.1 Übersicht

Nach einer kurzen Einleitung beginnt dieses Kapitel mit einer Erläuterung, was Multimedia, aus rechtlicher Sicht betrachtet, darstellt. Anschließend wird die Bedeutung des Urheberrechts und seine Funktion erläutert. Dabei geht es um den Gegenstand und den Schutzbereich des Urheberrechts und was im Sinne des Urheberrechts unter einem Werk verstanden wird. In diesem Zusammenhang werden die rechtlichen Pflichten und Lasten bei der Werkentstehung und Bearbeitung ebenso besprochen, wie die Rechte und Pflichten beim Zitieren, Parodieren, und Übernehmen. In einem weiteren Abschnitt werden Erwerb und Vergütung von Nutzungsrechten behandelt. Dabei spielen auch und gerade bei einer Multimedia-Anwendung sogenannte Verwertungsgesellschaften eine große Rolle. Hier werden einige davon vorgestellt. Zum Schluß gibt es noch ein paar Hinweise, wer einem Handreichungen und Hilfestellung bei der Vertragsgestaltung geben kann. Da der Rechtsbildungs- und Vereinheitlichungsprozeß im Internet noch im Werden ist, gehe ich nur in einem kurzen Abschnitt darauf ein, allerdings nicht ohne den Hinweis an dieser Stelle, die rechtliche Entwicklung scharf zu beobachten. Im "Serviceteil" gebe ich noch einige WWW-Adressen bekannt, über die ich im Laufe meiner Recherchen "gestolpert" bin

11.2 Einleitung

Es liegt mir fern, die versammelte Internetgemeinde kollektiv des Rechtsbruchs zu beschuldigen. Es ist jedoch, wie gerade bei den einschlägigen Pornografie- und Hetzschriften- Skandalen, die in jüngster Zeit die Weltöffentlichkeit erschütterten, zu erkennen, daß vielfach ein Rechts oder Unrechtsbewußtsein bei der Verwendung oder gar Aneignung fremden Schrift- und Bildgutes scheinbar oder auch tatsächlich kaum vorhanden ist. In der Öffentlichkeit entsteht dadurch sehr leicht der Eindruck, das Internet sei ein rechtsfreier Raum, in dem der Anarchie gegen alle herrschenden Systeme und Normen Tür und Tor geöffnet sind. Daß dem in der Realität aber ganz und gar nicht so ist, möchte ich in dem kurzen nunmehr nachfolgenden Artikel in wiederum kurzen, fast skizzenhaften Zügen darstellen. Die Eingangs erwähnten Skandale ließen in der Öffentlichkeit den Ruf nach einer staatlichen Kontrolle des Internet laut werden. Dabei geht es aber auch um die Frage, wer in einem solchen Fall die Verantwortung für die Verbreitung solchen Bild- und Schriftgutes trägt. Viele suchen in den Providern die Schuldigen zu sehen, doch diese können wiederum nur für ihre eigenen Server und Seiten haftbar gemacht werden, nicht zuletzt, weil die Telekom auch nicht für die mittels ihrer Telefonleitungen abgesprochenen Straftaten haften kann, da dies ein ständiges Abhören der Leitungen voraussetzen würde.

Zugegeben, ich versuche nur, die Lust auf "Mehr" zu wecken, doch das Thema ist sehr weitreichend, insbesondere, wenn das Schlagwort der "Globalisierung" mit ins Spiel kommt. Dies bezieht sich im Wesentlichen auf das ebenfalls zum Schlagwort gewordene Internet, dem weltweit umspannenden chaotisch organisierten Netzwerk von Millionen von Computerbenutzern, angeführt von Universitäten und Re-

chenzentren, den Knoten.

Die rechtliche Situation stellt sich denn auch in den unterschiedlichen, am Internet beteiligten Staaten differenziert dar. Ich betrachte lediglich den rechtlich vergleichsweise winzigen Abschnitt der Rechte und Gesetze, die die elektronische Publikation eines Werkes betreffen.

Die zugrundeliegenden Schriften und Artikel beziehen sich dabei hauptsächlich auf die CD-ROM, ein offline Medium, allerdings mit rasch steigender Popularität.

11.3 Multimedia

11.3.1 Was ist Multimedia - aus rechtlicher Perspektive

Allenthalben wird in den Medien viel über Multimedia und ihre Anwendungen, Wünsche und Ziele geredet. Das ZDF und auch andere Sender zeigen Beispiele und Anwendungen in eigens kreierten Sendungen wie Computerclub, Computerecke und anderen. Das geschieht unter vielerlei Voraussetzungen und in vielen verschiedenen Kontexten, je nachdem, von welcher Warte aus man an das Thema herangeht. Mein Blickpunkt betrifft die rechtlichen Aspekte der elektronischen Veröffentlichung, hat also mit der technischen Seite und der Machbarkeit von verschiedenen Dingen wie etwa der Herstellung einer Buch-CD oder den neuesten Glanzstücken in HTML nichts zu tun.

Doch an dieser Stelle sollte ein juristischer Fachmann einen Versuch unternehmen, den Begriff Multimedia zu erläutern [Bak96]:

“Multimedia“ hat sich zum umgangssprachlichen Sammelbegriff für die verschiedensten Erscheinungsformen und Anwendungsmöglichkeiten digitaler Informations- und Kommunikationstechnologien entwickelt, entbehrt aber gleichzeitig der sowohl für praktische wie wissenschaftliche Problemlösungen erforderlichen exakten Definition. Die bisweilen auch von den Euphorien des Zeitgeistes getragenen juristischen Beiträge zum “Thema“, das recht eigentlich so noch keines ist, gipfeln dann in Gleichungen wie “Multimedia [ist gleich] Multilegia“. Damit wird begrifflich die lateinisch so virtuos bekannteste Leere von der Technologie ins Recht übertragen.

[...]

“Multimedia“ ist [...] wesentlich gekennzeichnet durch die digitale Verarbeitung, Speicherung und Übertragung von Schrift-, Ton- und Bildinformationen in einer (digitalen) Einheitssprache, wodurch die unterschiedlichen Informationsströme und -arten in körperlicher wie unkörperlicher Form austauschbar, manipulierbar und beliebig mischbar werden.

[...]

Urheberrechtliche Bedeutung hat “Multimedia“, wenn es sich bei den digital übersetzten, vermischten und übertragenen “Informationen“ um bereits bestehende urheberrechtlich geschützte Werke oder Werkteile handelt.

Bakker [Bak96] versteht unter körperlicher und unkörperlicher Form die verschiedenen Medien, Möglichkeiten der Datenspeicherung und Vervielfältigung.

Ein Beispiel: Ein Buch ist eine körperliche Verbreitungsform, weshalb Drucken auch einen körperlichen Vervielfältigungsvorgang darstellt. Verschicke ich dieses Buch aber in Form von Datenbündeln ist dies eine unkörperliche Vervielfältigung, solange, bis dieses Buch wieder ausgedruckt wird.

Auf unserem Weg zum papierlosen Büro, von dem wir aber noch eine ganze Strecke entfernt sind, ist dabei unsere papierne Gesellschaft, denn ein Urhebergesetz für gedruckte Werke gilt schließlich bereits seit 1965, nur eine Zwischenstation. Die Juristen in vielen Ländern sind ständig bemüht, die Gesetzeslage dem technisch machbaren nachzuführen, haben jedoch auch ständig einen teilweise nicht unbeträchtlichen Rückstand aufzuholen. Dadurch können natürlich Situationen entstehen, in denen etwas technisch machbar ist, der rechtliche Schutz des Urhebers jedoch noch nicht gewährleistet werden kann.

Für einen Autor heißt es also stets: “Wie nehme ich meine Rechte wahr?”

Bei einer multimedialen Anwendung kann es aber schon einmal passieren, daß der Autor allein mit der Rechtswahrnehmung überfordert ist, insbesondere, wenn es sich bei den Erwerbern um viele Gesellschaften handelt, die eigentlich alle nur ein kleines Stück von ihm haben wollen. In vielen Sektoren der Publikation sind nun sogenannte Verwertungsgesellschaften entstanden, die einem Autor die persönliche Rechtswahrnehmung abnehmen und ihm somit das Leben vereinfachen.

Bei einer Multimedia- Veröffentlichung kann aber auch der Herausgeber oder Verlag ganz schön ins Schwitzen kommen. Für viele unterschiedliche Teile, aus denen eine solche Anwendung bestehen kann werden unterschiedliche Rechte benötigt, die jeweils wieder einzeln bei unter Umständen zig verschiedenen Verwertungsgesellschaften zu erwerben sind. Anstatt Lektoren mit der Duchsicht und Korrektur der Werke zu betrauen müssen immer größere Erwerbungs- und Rechtsabteilungen für die Verlage den Gesetzeswald im Blick behalten.

Woraus besteht Multimedia?

Es sind verschiedene Veröffentlichungs- und Darbietungsformen in einer einzigen Anwendung. Die Darbietungsformen können beispielsweise folgende sein:

- Text mit Bildern gekoppelt, wie es etwa bei Dokumentationen der Fall ist
- Text mit Videosequenzen. Kommt auch oft in Dokumentationen vor
- Videosquenzen mit Sprache: z.B. in URL: <http://WWW.tagesschau.de/>
- Interaktive Anwendungen: Lernprogramme, CBT-Anwendungen
- Das Buch nach Maß: Hier liegen einzelne Teile, Kapitel oder Artikel auf einem Server. Der Anwender stellt sich nun sein Buch selbst zusammen und entrichtet die Lizenzgebühr nach der Menge des verwendeten Materials

11.4 Das Urheberrecht

Ohne das Urheberrecht wäre unsere literarische Landschaft nicht lebensfähig, denn die Verlage hätten keine Veranlassung mehr, ihren Autoren für deren Manuskripte etwas zu bezahlen. Tatsächlich ist es auch trotz des Urheberrechts für einen Autor sehr schwierig, nur von den Tantiemen seiner Bücher zu leben. Doch was für Bücher gilt, gilt noch lange nicht für eine CD oder gar das Internet, wird sich so mancher sagen. Die Nachfolgende Definition zeigt aber, daß das Urheberrecht, das zum Schutze der Autoren vor dem Raub ihres geistigen Eigentums bestimmt ist, weitaus flexibler ist, als es allgemein von einem Gesetz angenommen wird.

11.4.1 Gegenstand und Schutzbereich des Urheberrechts

Hier eine Beschreibung der Leistung des Urheberrechts von Rainer Bakker [Bak96]:

Das Urheberrecht (*droit d'auteur*) nach kontinental-europäischem Verständnis ist das umfassende und eigentumsähnliche Recht des Werkschöpfers an seinem individuellen geistigen Werk. Schutzfähige Werke sind alle, aber auch nur "persönliche geistige Schöpfungen" (§2 Abs. 2 UrhG) sowie die verwandten Schutzrechte (§70 ff. UrhG). Schutzfähig ist danach nicht die Idee, sondern der geistige Inhalt, der sich in einer eigen-schöpferischen Formgebung konkretisiert. Das Urheberrecht entsteht *eo ipso* mit der „Schaffung des Werkes in der Person des Werkschöpfers. Es bedarf keiner Anmeldung oder Registrierung. 1994 ist die Schutzdauer durch eine Richtlinie der EU auf 70 Jahre *post mortem auctoris* vereinheitlicht worden.

Jede eigenständig verfaßte Arbeit wird also als eigenständiges Werk betrachtet, dessen uneingeschränktes und nicht übertragbares Recht beim Urheber selbst liegt. Um also etwas veröffentlichen zu können muß der Urheber einen zweiten Bestandteil des Urhebergesetzes nutzen, der die Vergabe und Inhalte von Nutzungsrechten regelt, wie Bakker weiter schreibt [Bak96]:

Das Urheberrecht hat nach §11 UrhG zwei Hauptbestandteile: es schützt den Urheber in seinen geistigen und persönlichen Beziehungen zum Werk (Urheberpersönlichkeitsrecht) und in der Nutzung des Werkes (Verwertungsrechte) in körperlicher und unkörperlicher Form. Das Urheberrecht selbst ist zu Lebzeiten des Urhebers nicht übertragbar. Der Urheber kann lediglich anderen Personen das Recht einräumen, sein Werk auf einzelne oder alle Nutzungsarten zu nutzen (Nutzungsrechte). Die Nutzungsrechte können räumlich, zeitlich oder inhaltlich beschränkt eingeräumt werden und ausschließlich (exklusiv) oder nicht ausschließlich (sog. einfaches Nutzungsrecht) begründet werden.

Die wichtigsten Rechte, die ein Urheber einem Nutzer übertragen kann sind das Vervielfältigungsrecht, das Verbreitungsrecht, das Vorführungsrecht und das Senderecht. Diese Rechte betreffen die Herausgabe eines Werkes unmittelbar und beziehen sich auf im Wesentlichen körperlichen, also materiellen, Verbreitungs und Kopierverfahren. Daraus begründet sich beispielsweise die Abgabe einer Gebühr des Kopiergerätebetreibers für jede auf seinen Geräten angefertigte Kopie eines Werkes.

In unserem digitalen und multimedialen Zeitalter mit der immer weiter zunehmenden Möglichkeit des Datentransfers durch Netzwerke taucht hinsichtlich der Verbreitungsformen ein Problem auf, da die nun immer stärker genutzten Verbreitungsverfahren für Informationen nunmehr den körperlichen Charakter verloren haben und als unkörperliche Verfahren bezeichnet werden können. Dies betrifft alle Werkformen, sei es nun Musik, Bücher, Video, Film und Bild. Diese Verbreitungsformen sind nur sehr schwer zu kontrollieren, da sie als unkörperliche Verbreitungsformen nicht unter das Verbreitungsrecht fallen. Hier könnte nur noch das Senderecht greifen, das auf ein ebenfalls unkörperliches Verbreitungsverfahren zielt. Bei einer Point to Point Übertragung von Daten ist aber der Adressat keineswegs die allgemeine Öffentlichkeit, so daß aus rechtlicher Sicht das Werk so nicht vor Mißbrauch geschützt ist. Abhilfe kann nur durch neue Gesetze oder die Anpassung der bereits bestehenden Gesetze an eine durch die neuen Medien verursachte veränderte Ausgangslage schaffen.

1994 ist die Schutzdauer durch eine EU-Richtlinie von in Deutschland 50 auf 70 Jahre nach dem Tode des Urhebers vereinheitlicht worden. Das deutsche Urheberrecht besteht seit 1965 und wurde und wird immer weiter ergänzt und angepaßt. Teilweise betreffen diese Anpassungen eine europäische, aber auch eine weiter ausgreifende internationale Übereinkunft, wie etwa obige EU-Richtlinie.

Das Urheberrecht hat nach §11 UrhG zwei Hauptbestandteile:

Es schützt den Urheber in seinen geistigen und persönlichen Beziehungen zum Werk (Urheberpersönlichkeitsrecht) und in der Nutzung des Werkes (Verwertungsrechte) in körperlicher und unkörperlicher Form. Das Urheberrecht selbst ist zu Lebzeiten des Urhebers nicht übertragbar. Der Urheber kann lediglich

anderen Personen das Recht einräumen, sein Werk auf einzelne oder alle Nutzungsarten zu nutzen (Nutzungsrechte). Die Nutzungsrechte können räumlich, zeitlich oder inhaltlich beschränkt eingeräumt werden und ausschließlich (exklusiv) oder nicht ausschließlich (sog. einfaches Nutzungsrecht) begründet werden.

11.4.2 Was ist alles geschützt?

- Text
- Bild
- Film / Video
- Musik / Sprache

11.4.3 Was ist ein Werk?

Was ist ein Werk? Diese Frage muß in letzter Zeit immer wieder neu gestellt werden. Wenn ich ein Buch in die Hand nehme, ist es eindeutigerweise ein Werk, nicht nur weil das Urhebergesetz es so definiert. Aber es sind ebenso wie das Buch oder der Zeitschriftenartikel Bilder, Computerprogramme, Filme oder Videos und Musikstücke als Werke zu betrachten und genießen somit den Schutz des Urhebergesetzes. Bei einer Multimedia-CD-ROM besteht der Schutz für die Gesamtkonzeption der aus vielen Einzelteilen bestehenden CD. Sie werden als geistiges Eigentum des Urhebers betrachtet und gelten somit als einzelnes Werk eines Autors.

Nach der Einführung einer allgemeinen Zugänglichkeit des Internet und des Aufbaus dezentraler Informationssysteme wie Gopher, WAIS und WWW geht die derzeitige Rechtsauffassung in die Richtung, auch WWW-Seiten als Werke zu betrachten und sie genießen somit selbstverständlich ebenfalls den Schutz des Urhebergesetzes.

11.5 Urheberrecht und digitale Werknutzung

Stichworte: Digitale Einspeicherung; Datenübertragung / Kopie; Zustimmungspflicht - wer und wann

Nur der Autor darf mit seinem geistigen Eigentum, dem Werk, alles anstellen, was ihm einfällt. Hingegen ist die nächste Instanz in der Kette einer elektronischen oder auch sonstigen Veröffentlichung, nämlich der Herausgeber, in seinen Handlungen mit diesem ihm vom Autor anvertrauten Werk sehr stark an das Urheberrecht gebunden. Möchte er die Datei des Werkes kopieren, so ist dies eine Vervielfältigung. Einer solchen Kopie seines Werkes muß der Autor und Urheber zustimmen. Die Zustimmung erfolgt im Verlagswesen durch geeignete Verträge, in denen der Verlag beispielsweise das Verbreitungsrecht erwirbt. Die Rechtsunsicherheit bei einer elektronischen Kopie entsteht durch die Art des Verfahrens, insbesondere durch das Verschicken von Daten über Netzwerke bleibt das Verfahren unkörperlich, bis die Datei am Zielort in einer Datenbank gespeichert oder ausgedruckt wird.

Ein solcher Kopiervorgang wäre aber auch das Aufrufen einer Datei durch beispielsweise eine Textverarbeitungssoftware. Hierbei wird von diesem Werk bereits eine Kopie im Hauptspeicher des aufrufenden Rechners angelegt.

Nach Art. 5 Abs. 1 des Entwurfs der EU-Datenbank-Richtlinie soll die Aufnahme von Werkdaten in eine Datenbank künftig generell der Zustimmung des Urhebers unterliegen.

Es sind somit der Verbreitung eines Werkes ohne die Zustimmung des Autors enge Grenzen gesetzt.

11.6 Der digitale Herausgeber / Werkbearbeitungen

Stichworte: Werkübernahme und Bearbeitung, Parodie, Zitat, Plagiat, gemeinfreie Werke

11.7 Übernahme und Veränderungen bestehender Werke

Stichworte: neue Verbreitungsmöglichkeiten; neue Kopiermöglichkeiten; neue Eingriffs- und Veränderungsmöglichkeiten; Manipulationen; Entstellung eines Werkes; Bearbeitung und Umgestaltung

Die digitale Werknutzung eröffnet nicht nur neue Verbreitungs- und Kopiermöglichkeiten, sondern auch Eingriffs- bzw. Veränderungsmöglichkeiten auf fremde Werke und Werkteile. Das Urheberrecht schützt den Urheber nicht nur vor Entstellungen und beeinträchtigenden Manipulationen seines Originalwerkes insgesamt, sondern auch vor unbefugter Entnahme/Nutzung von Werkteilen.

11.7.1 Die Bearbeitung und Umgestaltung vorbestehender Werke

Es handelt sich um ein selbständiges urheberrechtlich geschütztes Werk des Bearbeiters nach §3 UrhG, wenn es sich dabei um eine persönliche geistige Schöpfung des Bearbeiters handelt (wie etwa Übersetzungen und Dramatisierungen von Schriftwerken). Das entscheidende Kriterium ist dabei die Gestaltungshöhe des Vorlagewerkes. Je auffälliger die individuelle Eigenart des Vorlagewerkes (Gestaltungshöhe) ist, desto eher und stärker prägen dessen Eigenarten auch das bearbeitete Werk. Je geringer die Eigenprägung des Vorlagewerkes ausfällt, desto eher wird es in der Bearbeitung aufgehen. Sollte der Autor eines bearbeiteten Werkes sein eigenes Werk nicht mehr aus dem anderen Werk erkennen könne, so ist dies in der Gestaltungshöhe sehr weit oben anzusiedeln. Ist hingegen das Werk nur durch kopieren und Einsetzen zustande gekommen, so ist die Gestaltungshöhe eher niedrig einzustufen. Dabei richtet sich die Gestaltungshöhe nach dem Anteil des in die Bearbeitung eingeflossenen eigenen Gedankengutes des Bearbeiters. Aber auch die Stelle, an der der Bearbeiter sich selbst eingebracht hat, ist wichtig für die Ermittlung der Gestaltungshöhe. Die Veränderung einer Schlüsselstelle der Ursprungswerkes wiegt dabei weit schwerer, als eigene Ergänzungen an unwesentlicher Stelle. Richtwerte hierfür sind mir jedoch nicht bekannt.

Stellt die Veränderung des vorbestehenden Werkes noch keine persönliche geistige Schöpfung dar, so handelt es sich um eine bloße Umgestaltung, die ihrerseits keinen urheberrechtlichen Schutz genießt.

Für Bearbeitungen und Umgestaltungen gilt, daß beide nur mit Zustimmung des Urhebers des Originalwerkes/Vorlagewerkes veröffentlicht oder verwertet werden dürfen (§23 UrhG). Die bloße Herstellung der Bearbeitung oder Umgestaltung dagegen ist grundsätzlich frei. Das bedeutet, daß bis zum Veröffentlichungszeitpunkt hin alles mit einem schon vorbestehenden Werk angestellt werden darf, was literarisch oder technisch möglich ist. Hier greifen dann wieder Verbreitungs- und Senderechte, die vor einer Veröffentlichung durch den Ersturheber freigegeben werden müssen. Für die elektronische Veröffentlichung bedeutet dies, daß eine Zusammenstellung auf einer CD-ROM bereits eine Bearbeitung darstellt. Gleiches gilt für die Herausgabe von Büchern auf CD-ROM. Die interaktive Version eines Brockhaus-Lexikons wäre beispielsweise solch eine Bearbeitung.

11.7.2 Schutz vor Entstellungen

Wenn eine Bearbeitung veröffentlicht werden soll, kann der Urheber die Veröffentlichung verbieten, wenn er sein Werk durch die Bearbeitung entstellt (§14 UrhG). Bakker schreibt dazu: [Bak96]

Dieses Urheberpersönlichkeitsrecht bietet also Schutz vor Verzerrungen oder Verfälschungen der Wesenszüge eines Werkes, wie etwa der Streichung wesentlicher Werkteile, der

Verstümmelung oder gar der Sinnentstellung. Entstellungen können auch vorliegen, wenn die Wiedergabe unter herabsetzenden Begleitumständen oder an herabwürdigenden Orten geschieht.

Kabarettnummern fallen unter den Begriff der Parodie, auch wenn manchmal der Eindruck einer sinnentstellten Wiedergabe entstehen sollte.

11.7.3 Zitat

In jeder wissenschaftlichen Arbeit muß auch im rechtlichen Sinne korrekt zitiert werden, obwohl auch innerhalb der Wissenschaft verschiedene Zitierweisen bestehen, über die die jeweils andere wissenschaftliche Richtung geteilter Meinung sein kann. Rainer Bakker hat hier im rechtlichen Sinne eine allgemein gut zu verwendende Begriffsbestimmung gefunden, die hier kurz zitiert wird [Bak96]:

Ein Zitat im urheberrechtlichen Sinne liegt dann vor, wenn einzelne Stellen eines vorbestehenden Werkes in einem neuen, selbständigen Werk verwendet werden, wobei die Stellen aus dem vorbestehenden Werk nicht die Basis des neuen Werkes bilden dürfen, sondern in diesem lediglich “angeführt” werden. Die Rechte des Urhebers werden durch die in §51 UrhG gewährleistete Entlehnungsfreiheit (Zitierfreiheit) im Interesse der Allgemeinheit beschränkt. Die Zitierfreiheit besteht aber nur im Rahmen des Zitatszweckes. Dieser liegt in der Herbeiführung einer bestimmten fragmentarischen Assoziation. Die übernommenen Stellen dürfen also nicht als thematischer Einfall des neuen Werkes auftreten und nicht variiert werden. Ein prominentes Beispiel für ein Musikzitat ist das Anführen der “Marseillaise” im Beatles-Lied “All you need is love”, wodurch auf die Franzosen als ‘Erfinder’ der “l’amour” hingewiesen wird.

Wenn das “neue” Werk fast durchgängig aus Zitaten besteht, die nicht angegeben sind, dann spricht man von einem Plagiat, d. h. der Autor hat seinen Text bei einem anderen Autor entlehnt.

11.7.4 Das Plagiat

Als Plagiat bezeichnet man allgemein die bewußte Aneignung fremden Geistesgutes. Plagiator ist derjenige, der ein fremdes Werk oder Teile eines fremden Werkes als sein eigenes Werk ausgibt und somit “geistigen Diebstahl” begeht.

Der Plagiator begeht eine zivilrechtlich unerlaubte und zum Schadensersatz an den Autor verpflichtende Handlung, die gleichzeitig auch noch strafbar ist.

Der Plagiator ist also derjenige, der seinen Text wörtlich bei einem anderen Urheber abschreibt, ohne ihn korrekterweise zu zitieren, und anschließend dieses Werk als sein eigenes geistiges Eigentum ausgibt.

11.7.5 Die Parodie

Im Urheberrecht versteht man unter “Parodie” die antithematische Behandlung eines vorbestehenden Werkes wobei dessen Elemente zwar aufgegriffen, aber in einen anderen Kontext gestellt werden mit dem Ziel, die übernommenen Werkteile komisch oder satirisch wirken zu lassen. Die Parodie stellt eine “freie Benutzung” des Ausgangswerkes im Sinne von §24 Abs. 1 UrhG dar und ist bis zur Grenze der Entstellung auch ohne Zustimmung des ursprünglichen Werkes zulässig. citeBakker96

Dies ist bei Kabarettprogrammen oder satirischen Artikeln in Zeitungen, Zeitschriften oder Büchern der Fall. Dabei ist die Grenze zwischen Parodie und Entstellung als hauchdünner Faden zwischen diesen beiden Gruppen zu betrachten. Es könnte also das nächste Wort das Wort zuviel sein. Als Parodiebeispiel könnte in unserem Fall ein elektronisches Bildschirmschonerprogramm gelten, das unter dem Titel “Bill does Windows” sicher auf so manchem Computer für Freude gesorgt hat.

Eine Ausnahme von dieser Regel gilt für die Parodie von Musikwerken. §24 Abs. 2 UrhG enthält den Grundsatz des “starren Melodienschutzes”. Danach darf die Melodie eines Musikwerkes nicht erkennbar diesem entnommen und einem anderen Musikwerk zugrundegelegt werden. Die “freie Benutzung” im Musikbereich ist damit also ausgeschlossen und die musikalische Parodie ohne Zustimmung des Ursprungsurhebers verboten.

11.7.6 Gemeinfreie Werke

- Schutzfrist abgelaufen - 70 Jahre nach dem Tod des Urhebers
- Zur Zeit der Entstehung keinen Urheberschutz

Die Benutzung gemeinfreier Werke wird auch im Zeitalter der intensiven digitalen Werknutzung mit ihren neuen Nutzungsmöglichkeiten unproblematisch bleiben. Das ist möglicherweise die einzig sichere Prognose. Gemeinfreie Werke sind solche Werke, deren Schutzfrist 70 Jahre nach dem Tode des Urhebers abgelaufen ist und Werke, die aus einer Zeit stammen, in der es noch keinen gesetzlichen Urheberschutz gab. Die weitere Entwicklung der digitalen und multimedialen Werknutzung wird zeigen, ob sich das dargestellte konventionelle Instrumentarium zur Abwehr von Urheberrechtsverletzungen auch in der digitalen Zukunft als tauglich zum effektiven Schutz der Kulturschaffenden erweist. Prognosen sind hier heute ebenso riskant und unsicher wie bei der Beurteilung von “Multimedia” selbst. Das Projekt Gutenberg [Gut97] versucht, indem jemand solche Werke eingibt, gemeinfreie Werke der Internetgemeinde zugänglich zu machen.

11.8 Erwerb und Vergütung der Nutzungsrechte (Lizenzvergabe/Lizenzerwerb)

Die Vielzahl der bei digitalen Multimedia-Projekten betroffenen Urheberrechte macht den Erwerb sämtlicher Nutzungsrechte, das können je nach Umfang der digitalen Wort-, Bild- und Musikdatenbank mehrere tausende sein, bei den gegebenen Marktstrukturen nahezu unmöglich. Eine gewisse Erleichterung bringen hier die verschiedenen Verwertungsgesellschaften (VG).

In Deutschland haben die Rechteinhaber begonnen, sich auf freiwilliger Basis zu einem “Rechtepool” zusammenzuschließen. Als erster Schritt zu dem wünschenswerten “One-Stop-Shopping” der Zukunft haben die deutschen Verwertungsgesellschaften die “Clearingstelle Multimedia” (CMM) errichtet. Vieles spricht dafür, vom wirtschaftlichen Schutzzweck des Urheberrechts her Multimedia als eigene Nutzungsart anzusehen. Alt-Verträge erstrecken sich dann auf die digitale und multimediale Nutzung, wenn es sich im Zeitpunkt des Vertragsschlusses um eine bekannte Nutzungsart gehandelt hat. Für Multimedia erst etwa zu Beginn der 90er Jahre. Verträge, die vor diesem Zeitpunkt geschlossen wurden, erfassen daher die Nutzung im Rahmen von Multimedia nicht! In diesen Alt-Fällen ist die digitale und multimediale Nutzung zustimmungspflichtig und damit separat zu vergüten. Die angemessene und “marktgerechte” Vergütung für die gegenüber der analogen Werknutzung erheblich intensiveren digitalen Nutzungsmöglichkeiten wird der Markt selbst finden müssen. Das Urheberrecht hat dabei die Aufgabe die mögliche “Ausbeutung der Rohstofflieferanten” zu verhindern.

11.8.1 Verwertungsgesellschaften

Es gibt derzeit etwa 9 verschiedene Verwertungsgesellschaften. Sie nehmen urheberrechtliche Nutzungsrechte wahr, die durch den Einzelnen nicht selbst wahrgenommen werden können.

Während etwa bei einem Buchvertrag der Autor zwar selbst den Vertrag mit dem Verlag machen kann, so bleibt ihm dies mit jedem Betreiber eines Kopierladens versagt. Letztere führen pro Kopie einen Pauschalbetrag an die Verwertungsgesellschaften ab.

Von diesen 9 Verwertungsgesellschaften möchte ich nur zwei Beispiele nennen:

- VG-Wort, Textbereich
Die VG-Wort vertritt die Rechte der ihr angeschlossenen Autoren im Bereich Text. Sie verwaltet die Rechte nicht selbst sondern versteht sich als Mittler zwischen dem Autor als Lizenzgeber und Herausgebern und Verlagen als Lizenznehmern.
- GEMA, Musikbereich
Die GEMA ist die wohl bekannteste Verwertungsgesellschaft. Ihr Schriftzug prangt auf beinahe jeder CD oder LP, die produziert werden. Der Veranstalter hat bei jedem Konzert entsprechende GEMA-Gebühren abzuführen. Sie verwaltet die ihr von den Komponisten und Interpreten anvertrauten Rechte selbstständig.
- VG Wissenschaft
Die VG Wissenschaft betreut die Rechte wissenschaftlicher Autoren.
- VG Bild-Kunst
Die VG Bild-Kunst vertritt bildende Künstler in der Ausübung ihrer Urheberrechte, die für Gemälde ebenso gelten wie für Photographien oder Skulpturen.
- GVL (Gesellschaft zur Verwertung von Leistungsschutzrechten)

Allerdings bieten die großen deutschen Verwertungsgesellschaften (GEMA, VG-Wort, GVL und VG Bild-Kunst) traditionell unterschiedliche Leistungen an und nehmen unterschiedliche Autorenrechte wahr. Die GEMA beispielsweise verwaltet die Rechte ihrer Urheber selbst und kann sie auch selbst an den Multimedia-Produzenten gegen Entgelt weitergeben. Die VG-Wort dagegen kann die Anfragen nur an die Autoren weiterleiten.

11.9 Hilfen bei der Vertragsgestaltung bei Multimediaprojekten

Wer ein Multimedia-Projekt beginnt weiß zu Beginn gar nicht, was in so einen Vertrag zwischen ihm und den Autoren der einzelnen Teile alles an Bestimmungen und Pflichten hineingeschrieben werden muß. Hierbei kann der Börsenverein des deutschen Buchhandels Hilfestellung leisten.

Nach der Darstellung des Börsenvereins des deutschen Buchhandels ist in einem Vertrag zwischen Herausgeber und Autoren bei einer Multimedia-Anwendung eine komplizierte rechtliche Struktur eingetreten, dadurch, daß viele unterschiedliche Rechte benötigt werden.

Folgende Rechte werden u.a. für eine Multimedia-Anwendung benötigt:

- Vervielfältigungsrecht, §16 UrhG;
- Verbreitungsrecht, §17 UrhG;
- Vorführungsrecht, §19 UrhG;
- Senderecht, §20 UrhG;
- Recht auf Wiedergabe durch Bild- und Tonträger, §21 UrhG;
- Bearbeitungsrecht, §23 UrhG;
- Teilwerknutzung; Recht auf Werkverbindung, Leistungsschutzrechte der darbietenden Künstler u.a. §§73, 85 UrhG

“Multimedia“ wird durch kommunikationsvermittelnde Elemente und Interaktivität in elektronischer Form (CD-ROM, CD-Audio, CD-I u.a.) gekennzeichnet. Für eine Multimedia-Anwendung wird eine Vielzahl von Rechten benötigt, deren Einholung so kompliziert werden kann, daß die Rechtswahrnehmung nur durch Verwertungsgesellschaften praktikabel erscheint. Es wird angeraten, sich um eine solche Rechteübertragung in schriftlicher Form zu bemühen. Musterverträge sind zu finden in:

“Handreichung zum Abschluß von Lizenzverträgen über die Nutzung von Verlagswerken durch Online-datenbanken“, 1993

“Die Handreichung zum Abschluß von Verträgen über Verlagswerke mit Software und über CD-ROM“, 1994

Vertragliche Gestaltungen zu Multimedia-Anwendungen fehlen ebenso wie im Autorennormvertrag, der auch keine Regelungen zum elektronischen Publizieren vorsieht. Neue Musterverträge befinden sich in der Sammlung “Recht im Verlag”. Berücksichtigt werden die Verbreitungsformen Print, CD-ROM und Online. Musterverträge unter Berücksichtigung von Elektronischen Publikationen sind hier:

Mitarbeiter-/Beiträgervertrag für Sammel- und Reihenwerke (Autorenvertrag), Herausgebervertrag, Kurzfassung Verlagsrechte an Zeitschriften, Illustratorenvertrag, Listen von Illustratoren, Abdruck- / Übernahme-Lizenzvertrag auch für electronic publishing, Übersetzungslizenzvertrag.

11.10 Internet und Urheberrecht

Stichworte: Gültigkeit des Urheberrechts; keine eigene Rechtsbildung im Internet; Rechtswahrnehmung im Ursprungsland

Im Internet ist für den Einzelnen das Urheberrecht des Landes zuständig, in dessen Hoheitsbereich das Werk publiziert, d.h. ins Netz gebracht wird. Ebenso muß auch der “Leser” des Werkes das in diesem Lande gültige Urheberrecht beachten und darf nicht einfach wild draufloskopieren. Zitieren mit Quellenangabe im korrekten Zusammenhang ist jedoch immer erlaubt.

Eine eigene Rechtsbildung im Internet gibt es nicht. Die sogenannte “Netiquette“ ist lediglich eine Selbstbeschränkung der Benutzer dieses Netzwerkes, um einen freundlichen Umgang miteinander bemüht. Für Veröffentlichungen im Netz gilt das Urheberrecht des Ursprungslandes, also das der USA für dort eingespeiste Werke und das der Bundesrepublik Deutschland für hier Veröffentlichtes.

Die verschiedenen Verwertungsgesellschaften haben sich bereits Gedanken zum Rechtesalat im Internet

gemacht. Dies führt wohl sicherlich zur Gründung einer neuen Verwertungsgesellschaft für das Recht, im Internet zu publizieren.

11.11 Einige WWW-Adressen zum Thema Recht

Die nachfolgende Liste kann auch im WWW bestaunt werden. Sie könnte man unter das Motto stellen "From LINKS to RECHT". Ich bin sicher, daß diese kleine Auswahl nur einen winzigen Teil des international schon heute im Internet verfügbaren Materials darstellt. (Stand Januar 1997)

<http://www.logon.ch/dzwysig/ejpd/anhang2.htm>
<http://www.jura.uni-muenchen.de/Institute/>
<http://www.recht.de/HyperNews/get/forums/urheberrecht.html>
<http://ig.cs.tu-berlin.de/RV/EU/1122-42.txt>
<http://ri.jura.uni-sb.de/IfRI/jur-pc/OrComPro.HTM>
<http://www.uni-duesseldorf.de/WWW/Jura/ hoeren/aufsatz.htm>
<http://www2.echo.lu/legal/de/ipr/verleih/erwaeg.html>
<http://www.tu-freiberg.de/urz/mittheft/maerz93.html>
<http://libronet.freinet.de/Weinknecht/gesetz.htm>
<http://www2.echo.lu/legal/de/ipr/software/text.html>
http://www.et.fh-osnabrueck.de/fbe/pool/doku/soft/subsection3_1_4.html
<http://www.kanzlei.de/jurinfo>
<http://www.infosec.ch/artikel/literat/ltu.htm>
<http://pinguin.dunkel.de/DunkelListe/YRRECHT.html>
<http://www.jura.uni-sb.de/woche1.html>
<http://www.sima.ch/proj.gr/recht/>
<http://libronet.freinet.de/Weinknecht/selbst.htm>
http://libronet.freinet.de/Weinknecht_Medien/epprofil.htm
<http://fub46.fddi1.fu-berlin.de:8080/~hagemr/grundstd.html>
<http://www.logon.ch/dzwysig/ejpd/anhang1.htm>
<http://www.berlin.de/text/politik/senatsverwaltung/006.html>
<http://131.188.176.132/wire>
http://www.tu-berlin.de/~fomabln/w3v/name__e.htm
<http://pinguin.dunkel.de/DunkelListe/YRRECHTSANWALT>
<http://www.ad.or.at/text/731.htm>
<http://www.gema.de/publik/jahr96/mm1.html>
<http://pinguin.dunkel.de/DunkelListe/YGGEWERBLICHER-RECHTSCHUTZ>
<http://pinguin.dunkel.de/DunkelListe/YAANWALTSKANZLEI>
http://bunsen.hrz.uni-marburg.de/jura/geninfo/persoverz/alphabet/lessmann_herbert
<http://teflon.rz.uni-augsburg.de/~binder/legal.html>
<http://pinguin.dunkel.de/DunkelListe/YMMARKENRECHT>
<http://www.uni-sb.de/rewi/fb2/csle/csdpger.htm>
<http://pinguin.dunkel.de/DunkelListe/YUURHEBERRECHT>
<http://www.recht.de/HyperNews/get/forums/urheberrecht/5.html>
<http://www.recht.de/HyperNews/get/forums/urheberrecht/1.html>
<http://www.informatik.uni-oldenburg.de/Mitteilungen/news-regel.html>
<http://krum.rz.uni-mannheim.de/zki-www.html>
<http://www.hbz-nrw.de/mlist/inetbib/199510/19951025.html>
<http://www.netlaw.co.at/z.html>
<http://www.jura.uni-sb.de/internet/News.html>
<http://www.uni-paderborn.de/fachbereich/AG/agmadh/WWW/GI/VG-Wort/merkblatt.html>
<http://www.gi-ev.de/beiraete/vg-wort.html>
<http://www.fh-brandenburg.de/~cultura/documents1/kulturfoerderung/sozlwerk.htm>

<http://www.inm.ch/efb/adurhg.htm>
<http://www.gema.de/publik/sonder/verwertungsgesellschaften.html>
http://www.gema.de/publik/sonder/gema_sonderdruck.html
<http://www2.echo.lu/legal/en/hoerlic2.html>
<http://www2.echo.lu/legal/de/ipr/kabsat/text.html>
<http://flc.jk.uni-linz.ac.at/hillinger/PT.html>
<http://sunsite.informatik.rwth-aachen.de/Knowledge/germlaws/urhg/p26.html>
http://www.tfh-berlin.de/usr1/stud/1/s581061/public_html/Texte/Urheberrecht.html
<http://www.uni-mannheim.de/mateo/recht.html>
<http://libronet.freinet.de/Weinknecht/mmlaw.htm>
<http://www.tanznetz.de/thema/b016.html>
http://129.70.12.36:8080/netahtml/vlb_/reva/revainh.htm
<http://www.gema.de/gema/kurzdarstellung.html>
<http://www.ocg.or.at/ALLGE/ORG/INST/vam.html>
<http://www2.echo.lu/legal/en/950426/kaufmann.html>
<http://www.ins.net/offener-kanal-dortmund/infoheft/inforech.htm>
<http://www.informatik.uni-hamburg.de/gutenb/gutenb.html>

Ergänzende Literatur:

[Koc95, Hek93, RSD96, Fro94, PDRK97a, PDRK97b, Ble95]

11.12 Literatur

- [Bak96] Rainer Bakker. Multimedia und Urheberrecht. *HMD*, 188:99–107, 188 1996.
- [Ble95] Jens Bleuel. *Online Publizieren im Internet*. Ed. Ergon, Pfungstadt, Bensheim, 1995.
- [Fro94] Nordemann Fromm. *Urheberrecht - Kommentar zum Urheberrechtsgesetz und zum Urheberrechtswahrnehmungsgesetz*. 8. Aufl. Stuttgart, Berlin, Köln, 1994.
- [Gut97] Projekt Gutenberg. *Projekt Gutenberg Startseite*. URL: <http://www.informatik.uni-hamburg.de/gutenb/gutenb.html>, 28.1.1997.
- [Hek93] Dr. Harald G. Heker. Rechtsfragen der elektronischen Textkommunikation. *ZUM, Zeitschrift für Urheber- und Medienrecht*, 8/9:401–407, August/September 1993.
- [Koc95] Frank A. Koch. Software-Urheberschutz für Multimedia-Anwendungen. *GRUR Gewerblicher Rechtsschutz und Urheberrecht*, 7:459–469, Juli 1995.
- [PDRK97a] Dr. Jürgen Becker Prof. Dr. Reinhold Kreile. *Verwertungsgesellschaften - Aufgaben und Arbeitsweisen von Verwertungsgesellschaften*. URL: <http://www.gema.de/publik/sonder/verwertungsgesellschaften.html>, 22.1.1997.
- [PDRK97b] Dr. Jürgen Becker Prof. Dr. Reinhold Kreile. *Multimedia und die Praxis der Lizenzierung von Urheberrechten*. URL: <http://www.gema.de/publik/jahr96/mm.html>, 22.1.1997.
- [RSD96] M.A. R.A. Stephan Detjen. Rechtsprobleme internationaler Datennetze. *AFP, Zeitschrift für Medien- und Kommunikationsrecht*, 1:44–46, 1. Quartal 1996.



Kapitel 12

Computer Based Training

Jewgenij Tarnorutskiy

12.1 Übersicht

Der Begriff Computer Based Training CBT¹ (Computerunterstütztes Lernen, computerunterstützter Unterricht) steht, für ein sehr populär gewordenes Thema: Wie kann uns der Computer beim heutigen Bildungsbedarf helfen? Einige Aspekte dieses Problems werden in diesem Kapitel angesprochen.

12.2 Einleitung

Hatte man früher am Ende des 8. Volksschuljahres alles fürs Leben Notwendige gelernt, so macht sich heute die Forderung vom lebenslangen Lernen breit. Wollen wir mit der technischen Entwicklung Schritt halten, haben wir uns permanent weiterzubilden bzw. ständig Neues zu lernen. Es findet eine Verschmelzung von Aus- und Weiterbildung statt. Die hohe Innovationsrate der Wirtschaft führt zu einer immer schnelleren Entwertung der Basisausbildung und einem stetig wachsenden Bedarf an beruflicher Weiterbildung und Nachqualifizierung von Erwachsenen.

Für das Bildungswesen bedeutet dies einen grundlegenden Wandel der Aufgaben, der Qualifikation der Lehrer und Dozenten sowie der organisatorischen Strukturen. So kann zum Beispiel nur ein kleiner Teil der im Beruf stehenden Erwachsenen in zentralen Ganztagschulen weitergebildet werden. Die Mehrzahl braucht eine dezentralisierte, zeitunabhängige Ausbildung möglichst direkt am Arbeitsplatz. Auch das Lernen in der Freizeit und zu Hause muß das Bildungssystem der Zukunft ermöglichen. Experten gehen davon aus, daß dies alles nicht allein mit herkömmlichen Ausbildungsmethoden bewältigt werden kann, sondern nur im Verbund mit „neuen Medien“. Das Selbstlernen mit dialogfähigen Kommunikationssystemen spielt dabei eine bedeutende Rolle.

Computer Based Training in Form interaktiver Lernprogrammen ist eine der effektivsten Selbstlernmethoden. Niemand, der sich über moderne Weiterbildungskonzepte Gedanken machen muß, kommt heute ohne eine detaillierte Kenntnis des Mediums CBT aus.

12.3 CBT als Alternative

Daß ein Computer mit didaktisch gut gemachten Lernprogrammen für viele Ausbildungsziele eine Alternative zu den herkömmlichen Lernmethoden darstellt, ist seit vielen Jahren bekannt. Anfang der 60er Jahren erfuhr das Bildungswesen in den USA und später auch in Europa - nicht zuletzt als Folge des sogenannten „Sputnik-Schocks“- eine enorme Aufwertung. Ausgebildete Lehrer waren knapp und so erhielten die Argumente derjenigen, die das Bildungsniveau über die Nutzung einer sich rasant entwickelnden Computertechnologie anzuheben versprochen, schnell Aufwind. Computer sollten als Medium des Lehrens und Lernens die Versäumnisse der vorangegangenen Bildungspolitik ausgleichen das Image des öffentlichen Schulwesens aufhellen. In der Folge entstand mit Hilfe von überwiegend öffentlichen Geldern

¹Hiermit ist nicht Chicago Board of Trade - die größte Rohstoffbörse der Welt gemeint

Lernsoftware für nahezu alle Fachgebiete, Zielgruppen und Schulformen.

Warum die früheren Anwendungen zu keinem dauerhaften Erfolg wurden, liegt nicht am Prinzip von CBT, sondern an der damals vorhandenen Hard- und Software. Man mußte mit aufwendigen, stationären Großrechnern arbeiten, hatte mit langen Antwortzeiten zu kämpfen und als Gestaltungsmittel standen weder Graphik noch Farbe zur Verfügung. Es war daher nicht erstaunlich, daß die anfängliche Euphorie schnell einer begründeten Skepsis wich und der Ansatz schon einige Jahre später als weithin gescheitert galt.

Mit preiswerten und leistungsfähigen Personal Computern hat sich das Bild vollständig gewandelt. Der PC steht als Arbeitsgerät da, wo auch die Ausbildung erfolgen soll: am Arbeitsplatz; er steht jederzeit zur Verfügung, und er steht individuell zur Verfügung. Hinzu kommt, daß moderne Personal Computer auch technisch Hervorragendes bieten: sie liefern gestochen scharfe Farbgraphiken, haben ermüdungsarme flimmerfreie Bildschirme, besitzen enorme Speicherkapazitäten und können über genormte Schnittstellen andere Lernmedien wie beispielsweise Video-Bildplatten oder Audiogeräte ansteuern.

Damit lassen sich heute sehr attraktive und wirkungsvolle Lernprogramme entwickeln, die nicht nur kognitive, sondern auch affektive und vor allem sogar psychomotorische Lernziele erreichbar machen.

Im Gegensatz zu anderen Medien ist der Computer jedoch nicht nur ein ausgezeichnetes Lehrmedium, sondern auch ein Führungsmedium. Das grundsätzliche Problem beim mediengesteuerten Lernen ist ja, daß das Führungsmedium den menschlichen Lehrer nachbilden muß. Dieser Nachbildung sind naturgemäß bei jedem technischen Medien enge Grenzen gesetzt. Insbesondere scheiden alle Medien aus, die nicht dialogfähig sind. Das trifft für das Fachbuch ebenso zu wie für die programmierte Unterweisung, aber auch für alle nur darstellenden audiovisuellen Medien.

12.4 Varianten und Einsatzgebiete

CBT kann als Sammelbegriff für eine Vielzahl methodenkonzeptioneller Varianten verstanden werden, wobei jede mit eigenen Möglichkeiten und Grenzen verbunden ist. Sie werden grob in tutorielle und nichttutorielle Formen unterteilt. Im folgenden gehen wir auf einige der wichtigsten Varianten ein, wobei zwischen diesen sicherlich Überschneidungen und Mischformen möglich sind.

12.4.1 Tutorielle Formen: Programmierte Instruktionen und Übungen

CBT in tutorieller Form stellt mit anderen Worten einen apersonalen (aber dafür gedulden, ständig bereiten, immer gut gelaunten) Tutor dar, der über aktivierende Dialoge Informationen präsentiert, Aufgaben stellt, Antworten überprüft und kommentiert. Dazu werden nach einer orientierenden Einführung die Lerninhalte in miteinander verknüpften Einheiten angeboten und jeweils in kleinen Schritten interaktiv dargestellt. Den dargebotenen Informationen schließen sich jeweils Verständnisfragen an, deren Beantwortung durch die Lernsoftware analysiert wird. Als Ergebnis der Antwortanalyse erhält der Lerner eine Rückmeldung: je nach Ergebnis verzweigt die Lernsoftware zu neuen Lernschritten, oder sie bietet Lernhilfen an, die dem Lerner bei der Überwindung seiner Verständnisschwierigkeiten helfen sollen. Abbildung 12.1 zeigt allgemeine Reihenfolge einer Tutoriellen Unterweisung.

Viele Programme sehen zudem eine mehr oder weniger ausgeprägte Benutzersteuerung vor, daß heißt der Lerner kann innerhalb des vorgesehenen Rahmens selber entscheiden, welche Lerninhalte er als nächste bearbeiten möchte. Mit zunehmender Benutzersteuerung verliert das Tutorial an Direktivität und erlaubt ein entdeckendes Lernen im Sinne eines „Navigierens in informationellen Netzwerken“. Die Arbeitsergebnisse können vom System gespeichert und beim Bedarf rückgemeldet werden. Zum Abschluß des Programms kann ein Abschlußtest vorgesehen werden, in dem die vermittelten Lerninhalte geprüft werden.

Im Vordergrund der tutoriellen Unterweisung steht also die Vermittlung neuer Inhalte eines Fachgebietes einschließlich der Überprüfung des Lernerfolges. Sie scheint dabei insbesondere für die Vermittlung solcher Inhalte geeignet zu sein, die

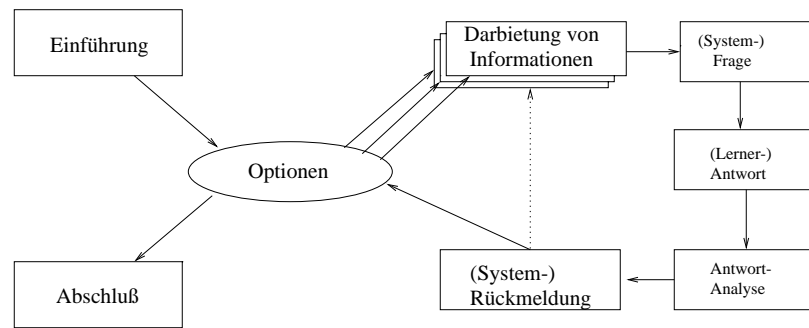


Abbildung 12.1: Tutorielle Unterweisung

- relativ eindeutig sind,
- nicht einer dauernden Veränderung unterliegen, d.h. relativ stabil sind,
- für einen größeren Anwenderkreis von Bedeutung sind.

Der Spektrum an möglichen Themen ist kaum eingrenzbar, wobei die meisten Anwendungen heute auf dem Bereich der betrieblichen Aus- und Weiterbildung zu treffen sind. Hier ein paar Anwendungsbeispiele:

Betriebswirtschaftliche Grundlagen: Das Spektrum an kommerziell angebotener bzw. selbst entwickelter Lernsoftware reicht von Grundwissen über betriebliche Funktionsbereiche (z.B. Marketing) über Vertiefungswissen zu betriebswirtschaftlichen Einzelfragen (z.B. Finanzplanung) bis hin zu Spezialwissen mit Branchenbezügen (z.B. Beleihungswertermittlung bei der Bank).

Informationstechnologische Grundlagen: In diese Kategorie fallen Programme über den Aufbau und die Funktionsweise eines Computers oder die Einführung in die Bedienung eines Betriebssystems oder Softwareproduktes.

Markt- und Produktinformationen: Einzelne Unternehmen gehen dazu über, ihre Mitarbeiter im Vertriebsbereich über Lernsoftware mit den jeweils aktuellen Markt- und Produktinformationen zu „versorgen“. So schulte etwa die Colonia-Versicherung ihre Außendienstmitarbeiter via Lernsoftware. Im Ergebnis wird von merklichen Umsatzsteigerungen berichtet.

Einsatz in der Sonder- und Förderpädagogik: Insbesondere für den Deutsch- und Mathematikunterricht im Bereich der Sonderschulpädagogik existiert eine Vielzahl von tutoriellen Programmen.

12.4.2 Nichttutorielle Formen: Computerunterstützte didaktische Simulationen und Spiele

Unter computerunterstützten didaktischen Simulationen und Spielen werden pädagogisch gestaltete Simulationen von Situationen verstanden, die zwar bestimmten Regeln gehorchen, aber durch relativ freie Aktivitäten veränderbar sind. Dieses Kriterium erfüllen am besten problemartig gestaltete, von den Lernenden manipulierbare Modellsituationen. Abbildung 12.2 zeigt allgemeine Reihenfolge eines Lernprogramms in nichttutorieller Form.

Beispiele für derartige Formen des Computereinsatzes sind

- in der berufsbezogenen Aus- und Weiterbildung: die bekannten Simulatoren von Flug- oder Fahrzeugen.
- in der Sonderpädagogik: Systeme zur Visualisierung von Sprache für Hörgeschädigte oder Gehörlose.

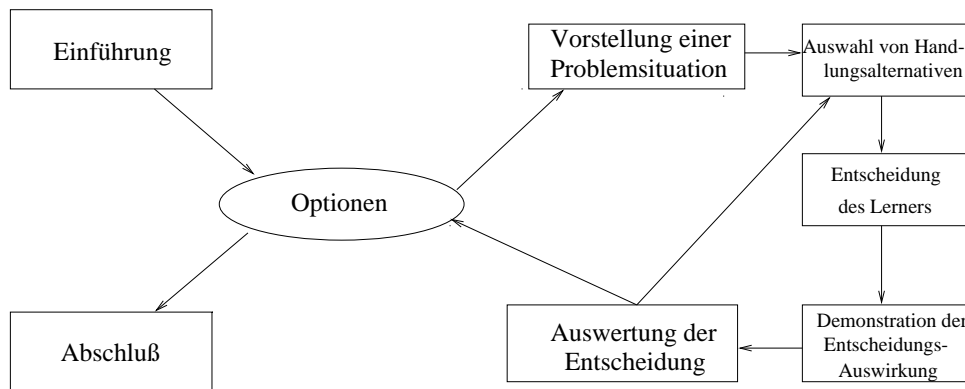


Abbildung 12.2: Nichttutorielle Formen

- im naturwissenschaftlichen Bereich: kostengünstiges und gefahrloses Üben bestimmter Methoden mittels simulierter Geräte und Anlagen.
- im Wirtschaftsunterricht: Unternehmens-, Börsen-, oder Wirtschaftsplanspiele zum Vermitteln neuer Einsichten in wirtschaftliche Systeme sowie Training von Entscheidungen.

Für die genaue Beschreibung dieser Lernformen verweisen wir auf [Eul92].

12.5 Intelligente tutorielle Lernsysteme

Schwäche tutorieller Systeme, wie zum Beispiel

- Begrenztheit der Lerninhalte auf die von einem Autor eingegebene Wissensbereiche,
- die Tatsache, daß der Lerner sich an die Software anpassen muß und nicht umgekehrt,
- reduzierte Form der Kommunikation zwischen Lerner und Lernsoftware

können durch Weiterentwicklungen teilweise überwunden werden. Ein Intelligentes Tutorielles Lernsystem zeichnet sich dadurch aus, daß es neben einer Wissenskomponente, in der die Wissensbasis des Lernstoffs in einer formalisierten, computergerechten Repräsentation vorhanden ist, über eine Diagnosekomponente verfügt, die im Dialog mit dem Lernenden über die Zeit hinweg zu einem immer adäquateren Bild seines jeweiligen Wissens führt. Der Vergleich von Lerninhalt und diagnostiziertem Wissensstand ermöglicht schließlich nicht nur den Abruf vorgefertigter Dialog-Elemente, sondern die Generierung neuer lerner- und situationadäquater Lernangebote. Die Realisation eines ITS sollte über folgende Komponente erfolgen.

- Wissenskomponente als Repräsentation des dem System zugrundeliegenden Gegenstandsbereichs (Expertenmodell).
- Schülerkomponente als Diagnostik zur Feststellung des aktuellen Wissenstandes des Lerners und seiner Veränderung während des Dialogs mit dem System.
- Lehrkomponente als Sammlung von Lehrangeboten und Lehrstrategien (zur Sequenzierung der Lernangebote in Abhängigkeit vom diagnostizierten Wissensstand des Lerners).
- Natursprachliches Dialogsystem zur Kommunikation mit dem Lerner.

Im idealen Fall wird vom Intelligenten Tutoriellen Lernsystem eine vollständige Simulation des menschlichen Lehrers erwartet. Die Konstruktion von einem lernfähigen Computer gilt als ein zentrales, wenngleich unerreichtes Ideal der Künstliche-Intelligenz-Forschung.

12.6 Hypertextsysteme

Eine weitere CBT-Form, die in der letzten Zeit insbesondere wegen der raschen Verbreitung von WWW-Browsern immer mehr an Bedeutung gewinnt, sind die Hypertextsysteme, in denen die Aneignung von Informationen nichtlinear erfolgt, d.h. man kann sich den Stoff in beliebiger Reihenfolge anschauen, indem man jeweils den Gegenstand aussucht, den man als nächstes sehen will.

Hypertextsysteme verzichten in der Regel auf tutorielle Dialoge und Lernkontrollen. Sie haben eine mit Enzyklopädien vergleichbare Funktion, die dazu dient, große Mengen von Wissen zu speichern und sie entsprechend den Kriterien des Benutzers zugänglich zu machen. Die Idee hinter Hypertextsystemen ist die, daß Menschen assoziativ denken. Dementsprechend werden die Informationen durch Assoziationen in Form von Querverweisen verbunden.

Die Vernetzungen und Assoziationen werden im Text explizit dargestellt, so daß man eine Oberfläche erhält, die dem menschlichen Denken nahekommt.

Einer herkömmlichen Enzyklopädie gegenüber hat der Computer große Vorteile:

- Das „Nachschlagen“ erfolgt automatisch, d.h. der Computer springt direkt zu der Stelle, die gewählt wurde.
- Geht man in einem Buch verschiedenen Verweisen nach, so weiß man bald nicht, wo die Suche begonnen hat. Der Computer merkt sich den Weg, man kann ihn jederzeit zurückverfolgen.
- Informationen bestehen nicht nur aus Text- und Bildseiten, sondern aus allen möglichen elektronisch gespeicherten Daten, auch aus Filmen, Sprachen, Musik oder auch aus ganzen Datenbanken.

Hypertextsysteme befreien von dem sturen „programmierten Unterricht“. Sie lassen dem Benutzer Gelegenheit zum *individuellen* Lernen. Jeder Student kann sich seinen eigenen Weg durch das Wissen bahnen. Dabei findet er seine eigenen Assoziationen und kann sich die benötigten Informationen aus großen Wissensbeständen selbst erschließen. Ihm wird eine Lernumgebung zur Verfügung gestellt, in der er das, was gelernt werden soll, im Sinne eines spielerisch-entdeckenden Lernens selbständig ausspürt, ausprobiert und anwendet.

Die Verantwortung bei der Auseinandersetzung mit solchen großen Wissensmengen kann natürlich nicht vollständig auf den Benutzer übertragen werden. Zum einen können Schwierigkeiten auftreten, die gewünschte oder benötigte Information in der Stofffülle überhaupt zu finden, zum anderen kann man bei einer großen Anzahl von Informationen schnell den Überblick verlieren. Unnötige Lernumwege und ein wenig zielgerichteter Lernvorgang sind die Folgen(„lost in hyperspace“ heißt das inzwischen). Doch hier kann und muß von Autoren von Hypertextsystemen Vorsorge getroffen werden. Diese meisten Systeme sind so angelegt, daß der Benutzer auf drei Arten nach Informationen suchen kann:

- durch Aktivierung markierter Startpunkte oder Tasten im Text folgt er spontan und intuitiv den Verbindungen;
- er kann sich aber auch vom System einen Weg vorschlagen lassen, von dem er allerdings jederzeit abweichen kann. Natürlich kann der Weg auch zwingend vorgeschrieben sein;
- es kann nach Schlüsselwörtern oder beliebigen Textstellen gesucht werden.

Eine weitere Dimension eröffnen Hypertextsysteme, indem sie neben der Abrufbarkeit des Wissens auch das Einbringen neuer Informationen, eigener Randnotizen, Assoziationen und Querverweise erlauben. So wie ein Student ein Lehrbuch für sich nutzbar macht, indem er es mit Anstreichungen und Randnotizen versieht, kann er sich mit einem Hypertextsystem aus einem großen Informationsbestand seine ganz persönliche Wissensbasis zusammenstellen und für künftige Informationen zugänglich machen.

12.7 Werkzeuge für Entwicklung von Lernsoftware

Prinzipiell können Computerunterstützte Lernprogramme, wie andere Anwenderprogramme auch, in jeder gängigen Programmiersprache geschrieben werden. Es gibt jedoch Werkzeuge, die die Erstellung

derartiger Programmen erheblich erleichtern. Für diese Werkzeuge hat sich der Begriff Autorensysteme durchgesetzt. Das Wesen der Autorensysteme besteht darin, daß sie Bausteine zur Verfügung stellen, mit denen bestimmte, in Lernprogrammen vorkommende Strukturen beschrieben werden können. Das reduziert nicht nur die Entwicklungszeit, sondern hat vor allem den Vorteil, daß Autorensysteme sehr schnell erlernbar sind. Schon nach kurzer Einarbeitungszeit können auch computerunerfahrene Autoren Lernprogramme erstellen. Dadurch kommen nicht nur programmiererfahrene EDV-Spezialisten als Autoren von Lernprogrammen in Betracht, sondern vor allem pädagogisch und didaktisch erfahrene Lehrer und Ausbilder, bei denen Programmierkenntnisse nicht unbedingt vorausgesetzt werden können.

Obwohl es mit Autorensystemen viel einfacher geworden ist, Lernprogramme zu erstellen, werden an die Autoren hohe didaktische Anforderungen gestellt; denn die Lernziele und der Lernstoff sind in eine für den Adressaten adäquate Lernstruktur zu bringen und in eine auf dem Computer ablauffähige Struktur zu übertragen. Dabei sind die Einzelheiten des Mediums zu beachten.

12.8 Möglichkeiten und Grenzen interaktiver Medien in der betrieblichen Bildung

Obwohl es nach Meinung vieler Experten keine Themen gibt, für die die interaktiven Lernmedien absolut ungeeignet wären, wird die Nutzung des Computers als Lernmedium heutzutage vor allem im Bereich der betrieblichen Aus- und Weiterbildung diskutiert und durchgesetzt. Nach der seit ein Paar Jahren laufenden Phasen der Pilotprojekte hat der gezielte Einsatz auf breiter Front begonnen. Folgende Kriterien sind für die Beschreibung der Möglichkeiten und Grenzen des Einsatzes interaktiver Medien von entscheidender Bedeutung:

- Adressaten
- Lerninhalte
- Lernziele
- Wirtschaftlichkeit
- Lernsituation

Auf diese Kriterien gehen wir genauer ein.

12.8.1 Adressaten

Grundsätzlich kann man beim Erstellen eines Lernprogramms nicht davon ausgehen, daß die Lernenden bereits Erfahrungen im Umgang mit Computer-Lernprogrammen besitzen. Mangelnde Erfahrung mit interaktivem Lernen darf nicht als genereller Hinderungsgrund für den Einsatz interaktiver Medien als Selbstlernmedien aufgefaßt werden. Die Lernprogramme müssen so leicht zu bedienen sein, daß man keine längeren Erweisungen benötigt, um mit ihnen zurechtzukommen.

Viel wichtiger als die Praxiserfahrung ist die Lernmotivation der Adressaten, also ihre Bereitschaft zum Lernen mit dem Computer, die bereits vor der Bearbeitung eines Lernprogrammes besteht oder aufgebaut werden muß.

Den Lernenden zum Lernen am Computer zu motivieren sowie seine eventuelle Ängste möglichst abzubauen ist eine der wichtigsten Aufgaben der Gestaltung von Lernprogrammen.

12.8.2 Lerninhalte

Unter dem Begriff Lerninhalt verstehen wir hier nicht, was, sondern vielmehr, wie detailliert der Lernstoff vorgestellt wird. So reicht häufig die Vermittlung von 25 % der technischen Möglichkeiten einer Software, um den Wissensbedarf von 50 % der Adressaten abzudecken.

Dazu ein Beispiel:

Um mit Hilfe eines Textverarbeitungssystems einfache Briefe bzw. sonstige Texte abzutippen und auszudrucken, muß man sicher nicht mehr als 25 % des Befehlsrepertoires beherrschen - ehe weniger. Wenn es stimmt, daß man damit schon die Anforderungen von 50 % der Anwender befriedigt, muß man damit sich natürlich fragen, ob es sich in einem Computer-Lernprogramm lohnt, mehr als diese 25 % der technischen Möglichkeiten zu erklären, in Anwendungsbeispielen zu üben usw.. Denn unabhängig von der absoluten Anzahl der Adressaten wird man mit dem doppelten Aufwand, wenn also 50 % der technischen Möglichkeiten erklärt werden, nicht einmal halbsoviele Adressaten erreichen wie zuvor - und erst ein gegenüber der Ausgangssituation vervierfacher Aufwand ermöglicht, die letzten 25 % der Adressaten zu erreichen.

Die richtige Wahl der Lernstofftiefe ist also sehr wichtig für den Aufwand und damit die Wirtschaftlichkeit.

12.8.3 Lernziele/Lerntechniken

Ohne Frage ist CBT von seinen Möglichkeiten her prädestiniert für kognitive Lernziele, also für Vermittlung von Wissen. Deswegen kann theoretischer Unterricht in der Regel völlig problemlos durch CBT ersetzt werden. Die visuellen Möglichkeiten der Lehrstoff-Darbietung auf dem Bildschirm mit Text, statischer und bewegter Farbgraphik und Simulation gehen dabei über das hinaus, was normalerweise im Live-Unterricht geboten werden kann. Nimmt man die mit dem PC mögliche Sprachausgabe und den Medienverbund mit Printmedien, CD-ROM und Video hinzu, so gibt es kaum ein kognitives Lernziel, das nicht erreichbar wäre.

Mit affektiven Lernzielen dagegen, also der Beeinflussung und Veränderung von Verhalten oder Einstellungen, tut sich der CBT- Autor erheblich schwerer als der Lehrer im Unterricht oder Seminar. Denn bei CBT fehlt sozusagen der Pulsschlag des Lebendigen, es fehlt die Suggestivkraft des unmittelbar anwesenden Menschen mit seiner Körperlichkeit, seiner Gestik, Mimik und Sprache. Das gilt natürlich nicht nur für CBT, sondern für jedes apersonale Medium. Zwar kann man durch ein besonders aufwendige Machart des Lernprogramms einiges ausgleichen, genau wie bei einer Druckschrift, die affektive Ziele verfolgt. Dennoch muß festgestellt werden, daß komplexe affektive Lernziele von CBT wohl nur im Medienverbund mit Video zu erreichen sind. Immer dann also, wenn die rein sachliche Information und Ausbildung übergeht in ausgeprägt werbliche Kundeninformation, imagebildende Verkaufsförderung, messewirksame Unternehmenspräsentation und ähnliche Ziele, so wird man sehr genau abwägen müssen, ob CBT nicht mit einer Bildplatte gekoppelt werden muß.

Noch ungünstiger sieht es bei psychomotorischen Lernzielen aus, also der Schulung und des Drills von praktischen Fertigkeiten. Denn diese Art von Lernziel ist bekanntlich nur durch eigenes Tun und vielfaches Üben zu erreichen. Bei der schulischen Ausbildung geschieht dies im praktischen Unterricht oder Workshop. Dabei erläutert der Lehrer zuerst die zu erlernende Fertigkeiten, führt sie gegebenenfalls auch selbst beispielhaft aus und läßt dann die Lerner üben. Der Kern der Sache ist, daß der Lerner die Fertigkeit an oder mit dem realen Gerät, der realen Maschine, usw. und auch mit den realen Werkzeugen erlernt und übt. Vor allem überwacht der Lehrer die Arbeit, korrigiert sie, gibt Lösungshinweise und motiviert den Lerner.

Auf der anderen Seite verfügt CBT über das einzigartige Mittel der Simulation. Damit läßt sich zwar eine echte praktische Ausbildung nur in wenigen Fällen ersetzen, fast immer jedoch drastisch verkürzen. Der Vorteil der Simulation ist, daß jeder „Handgriff“ vom Programm kontrolliert und kommentiert werden kann, ganz ähnlich, wie es einem Lehrer im praktischen Unterricht möglich ist.

12.8.4 Wirtschaftlichkeit

Wichtigste Faktoren der Wirtschaftlichkeit interaktiver Medien sind:

- Anzahl der Adressaten
- Dauer der Ausbildung (Lernzeit)

- Arbeitskosten der Adressaten
- Arbeitskosten der Ausbilder
- Hardware-Kosten

Die Anzahl der Adressaten pro Maßnahme ist natürlich besonders wichtig, deshalb steht sie auch an der ersten Stelle. Man muß aber bedenken daß es auf die absolute Zahl ankommt: d.h. 100 Adressaten pro Jahr bei einer Laufzeit des Programms von 4 Jahren ergibt die Gesamtzahl von 400 Adressaten - ebenso wie 400 Adressaten bei einer Laufzeit von nur einem Jahr.

Fast ebenso großen Einfluß nehmen jedoch auch:

- Die Dauer der Ausbildung, d.h. die benötigte Lernzeit - wobei man sich vom Einsatz interaktiver Medien generell eine Verringerung um mindestens ein Drittel erwarten darf,
- die Arbeitskosten von Lernern und Ausbildern - also nicht nur die reinen Lohnkosten, sondern bei den Lernern auch sonstige Ausfallkosten, wie Umsatzeinbußen etc., bei den Ausbildern die anteiligen Gemeinkosten pro Arbeitsplatz,
- und nicht zuletzt die Hardware-Kosten bzw. allgemein gesprochen die Kosten, die entstehen, wenn zusätzliche Geräte für die Ausbildung mit interaktiven Medien angeschafft werden müssen und nicht bereits im Unternehmen zur Verfügung stehen.

Es ist sehr wichtig, bei Wirtschaftlichkeitsberechnungen darauf zu achten, daß nicht Äpfel mit Birnen verglichen werden:

- Wer ein bereits existierendes Seminar, das nicht mehr neu konzipiert und produziert werden muß, mit einem neu zu entwickelnden CBT-Programm vergleicht, wird es schwer haben, die Wirtschaftlichkeit von CBT-Projekten zu begründen;
- Wer die Kosten für 10 neu zu beschaffende PCs zu Lernzwecken in ein einziges Projekt hineinrechnet, wird die gleichen Schwierigkeiten haben; und schließlich:
- Wer die Arbeitskosten des eigenen Ausbildungspersonals nicht oder zu niedrig berücksichtigt (etwa nur mit Lohnkosten, statt mit Lohnkosten plus anteiligen Gemeinkosten), wird nie einen realistischen Rentabilitätsvergleich durchführen können.

Alle diese Größen müssen in ein Rechenmodell zur Wirtschaftlichkeitsberechnung des Einsatzes interaktiver Medien Eingang finden.

12.8.5 Lernsituation

Haupteinsatzgebiet interaktiver Medien sollte von Anfang an das individuelle Lernen am Arbeitsplatz sein. Wirtschaftlichkeitsrechnungen aus der Praxis belegen, daß sich der Kostenaufwand - eine angemessene Adressatenzahl vorausgesetzt - gegenüber konventionellen Vermittlungsformen (Seminar) verringern läßt.

Dabei wurde (und wird zum Teil noch) vor allem an das Lernen am Arbeitsplatz gedacht, also das Lernen während der Arbeitszeit. Damit konnten zum Beispiel unproduktive, arbeitstechnisch bedingte Pausen sinnvoll überbrückt werden, so daß sich der Lerner den Stoff sozusagen nebenbei aneignet und aufs Ganze gesehen zumindest ein erheblicher Teil des Zeitaufwandes bei konventionellen Vermittlungsformen eingespart würde. Inzwischen hat sich erwiesen, daß das Lernen am Arbeitsplatz in vielen Fällen nicht allzu wörtlich genommen werden darf: technisch bedingte Leerzeiten von vorher unbestimmter Dauer eignen sich nicht gut für konzentriertes Lernen; heufig ist auch das Lernumfeld (Lärm, Störungen etc.) der Aufnahmefähigkeit nicht gerade förderlich.

Überlegungen wie diese machen deutlich, daß auch die Frage der Lernsituation, der gegebenen oder der aus didaktischen Gründen zwingend zu schaffenden, ein wichtiges Entscheidungskriterium bei der Beurteilung von Möglichkeiten und Grenzen des Einsatzes interaktiver Medien darstellt.

12.8.6 Schlußbetrachtungen

Zum Schluß wollte ich noch drei Grundtendenzen auflisten, die von CBT-Spezialisten bei dem zunehmenden Einsatz interaktiver Medien beobachtet werden.

1. Interaktive Medien erweisen sich häufig im Rahmen Kosten-Nutzen-Analysen als effizient. Dies gilt - trotz des nicht unbeträchtlichen Entwicklungsaufwands - vor allem für Ausbildungsmaßnahmen, die mit personaler Schulung bei größeren Adressatengruppen häufig kaum möglich sind. Beispiele dafür sind die Aufgaben der öffentlichen Verwaltung in den neuen Bundesländern oder die in vielen Unternehmen bevorstehenden Neuerungen im Zusammenhang mit der Einführung des europäischen Binnenmarktes.
2. Der Einsatz interaktiver Medien wird zunehmend weniger mit konkreten Einsparungen begründet als vielmehr mit der Notwendigkeit, zusätzliche Bildungsmaßnahmen durchzuführen, die anderweitig schon aus Kapazitätsgründen nicht realisiert werden können.
3. Das Lernen mit interaktiven Medien wird zunehmend als ein wichtiger Bestandteil erwachsenengerechter Formen des Lernens eingeschätzt. Das gilt nicht für autonome sondern auch für kooperative Formen der Selbstqualifizierung.

Weiteres Über verschiedene Aspekte des Einsatzes von CBT in betrieblichen Bildung findet man in [Sei93] und [Ste89].

12.9 Neues Lernen im Netz

Sicherlich können Internet oder damit verbundene Software nicht als Lernprogrammen aufgefaßt werden. Die Möglichkeiten des Netzes sind aber so groß, daß sie auch beim computerunterstützten Lernen nicht vernachlässigt werden dürfen. Internet hat sich bereits zur größten internationalen Kulturszene und der umfassendsten Beratungs- und Informationsbörse der Welt entwickelt. Die Bildungsangebote im Netz sind beinahe unbegrenzt. Die *Encyclopaedia Britannica* läßt sich genauso auf dem Bildschirm holen wie der Koran. Über das Netz führt auch ein Weg zu akademischen Weichen. Einige Universitäten in USA bieten ihren Studenten die Möglichkeit zu Hause via Internet zu studieren. Im Sommersemester 1995 verfolgten per Videokonferenz Studenten in Freiburg die Vorlesung, die ein Professor in Karlsruhe hielt [Kle86].

Durch die Entwicklung von neuer Netzsoftware z.B. Javasprache ist es möglich geworden Lernsoftware durch Internet zu nutzen.

12.10 Literatur

- [Eul92] Dieter Euler. *Didaktik des computerunterstützten Lernens. Praktische Gestaltung und theoretische Grundlagen*. 1992.
- [Kle86] Robert Kleinschroth. *Neues Lernen mit dem Computer*. Verlag rororo Computer Ludwig Ross, 1986.
- [Sei93] Christoph Seidel. *Computer Based Training*. Verlag für angewandte Psychologie, 1993.
- [Ste89] Hubert Steppi. *Computer Based Training Planung Design und Entwicklung interaktiver Lernprogramme*. Ernst Klett Verlag, 1989.



Kapitel 13

Kiosksysteme

Olfa Karboul

13.1 Übersicht

In diesem Kapitel wird das Kiosksystem vorgestellt. Es ist ein öffentlich zugängliches, rechnerbasiertes System, bei dem ein Benutzer für eine kurze Zeit Informationen abrufen und Transaktionen veranlaßt. Die folgenden Abschnitte erläutern den allgemeinen Aufbau eines Kiosksystems. Es wird dabei auf die Architektur, die Anwendungsbereiche, die Multimediale Kiosksysteme und die Bewertung eingegangen.

13.2 Einleitung

13.2.1 Was ist ein Kiosksystem?

Ein Kiosk, in der islamischen Baukunst, ist ein zentral gelegenes Gartenhaus auf rechteckigem oder polygonalem Grundriß mit Säulenvorhalle oder rings umgeben von einem nach außen offenen Bogengang. Heute ist das Kiosk ein Verkaufshäuschen u.a. für Zeitungen, Getränke, Süßigkeiten und ist meistens an belebten Straßen, Plätzen und Ausflugszielen zu finden. [Mey75]

In der Informatik versteht man unter einem Kiosksystem ein rechnergestütztes Informationssystem an öffentlich zugänglichen Orten. Über eine einfache Benutzungsschnittstelle, können überwiegend im Stehen und innerhalb einer relativ kurzen Verweildauer, von häufig wechselnden und meist unbekannten Benutzern, Informationen abgerufen oder Transaktionen ausgelöst werden.

13.2.2 Charakteristiken

Kiosksysteme werden an allgemein zugänglichen Orten aufgestellt, um Informationen für eine breite Öffentlichkeit anbieten zu können. Man muß davon ausgehen, daß die Benutzer keine Erfahrung mit Computer haben. Deshalb ist eine leicht verständliche und einfach zu bedienende Benutzungsschnittstelle notwendig. Dabei sollten auch körperliche Unterschiede beachtet werden. Man kann auch bemerken, daß eine einfache Benutzungsoberfläche die Akzeptanz und Bedienungsfreundlichkeit eines Kiosksystems positiv beeinflusst. Ein Computerlaie sollte nicht mit einer komplizierten Tastatur oder der Benutzung einer Maus konfrontiert werden. Deshalb sind die meisten Kiosksysteme mit einem sogenannten Touch-Screen (berührungsempfindlicher Monitor) oder mit einer einfach zu bedienenden Tastatur, die robust gebaut ist, ausgestattet.

Ein weiterer Aspekt ist die Individualität eines Kiosksystems, die nicht nur die Ein- und Ausgabe-mechanismen sondern auch das Design der Anwendung und vor allem das des Kioskgehäuses betrifft: Je nach Einsatzgebiet, Standort, angebotene Dienste und Zielgruppe unterscheiden sich die Systeme erheblich. Zum Beispiel benötigt das eine Möglichkeit zur Texteingabe, ein zweites dagegen einen Kartenleser und eine numerische Tastatur. Ein weiteres soll im Freien eingesetzt werden und benötigt deshalb ein schützendes Spezialgehäuse.

13.2.3 Klassifikation

13.2.3.1 Interaktionsgrad

Je nachdem, welche Einflußmöglichkeiten ein Benutzer auf die ihm präsentierten Informationsinhalte und den Informationsfluß hat, und abhängig davon, ob ein Kiosksystem Transaktionen durchführen kann oder nicht, sind folgende Kiosksysteme zu unterscheiden:

- Animationskiosk: der Benutzer hat keine Möglichkeit, (außer evtl. Start und Stop) Art, Umfang oder Reihenfolge der ihm präsentierten Informationen zu beeinflussen.
- Interaktionskiosk: Der Benutzer kann Art, Umfang und Reihenfolge der ihm präsentierten Informationen interaktiv beeinflussen.
- Transaktionskiosk: Der Benutzer kann durch seine Interaktion im System gehaltene Daten manipulieren (eine Transaktion auslösen).

Innerhalb dieser Beeinflussungsgrade können zum Teil noch weitere Unterschiede gemacht werden. Ein Interaktionskiosk kann Selektionskriterien explizit über eine Tastatur oder ähnliches einlesen, d.h. über gezielte Suchbegriffe oder komplexere Suchanfragen gesteuert werden oder über sogenannte Hyperlinks. Das sind interne Verzweigungen zu anderen Informationen, die dem Benutzer das Navigieren ermöglichen.

13.2.3.2 Verteilungsgrad

Bei der Informationsspeicherung, wie auch der Transaktionsverarbeitung kann zwischen lokalen und verteilten Kiosksystemen unterschieden werden.

- Lokale Systeme halten ihre gesamten Daten lokal (z.B. auf Festplatte oder CD-Rom-Laufwerken, die über den lokalen Datenbus an das System angeschlossen sind).
- Verteilte Kiosksysteme basieren auf einer verteilten Architektur, in der die Daten auf einem oder mehreren Servern verteilt gehalten werden. Dies ist vor allem bei der Nutzung multimedialer Daten (hohes Datenvolumen) von Vorteil, da nicht in jedem lokalen Kiosksystem mehrere hundert Megabyte Daten redundant gehalten werden müssen. Somit können Kosten für periphere Speichermedien eingespart und die konsistente Aktualisierung und Pflege des Datenbestandes wesentlich vereinfacht werden.

13.2.3.3 Unterstützte Medien

Man unterscheidet zwischen multimedialfähigen Kiosken, die kontinuierliche Medien unterstützen und nicht multimedialfähigen Kiosken, die ausschließlich diskrete Medien unterstützen.

- Multimediale Kiosksysteme unterstützen neben diskreten Medien wie Texte und Grafik auch kontinuierliche Medien wie Audio und Video.
- Text- und grafikbasierte Kiosksysteme unterstützen nur diskrete Medien (Text und Grafik).

13.3 Architektur und Bausteine eines Kiosksystems

Die Module eines Kiosksystems umfassen die graphische Benutzungsschnittstelle, die Systemsteuerung (Engine), Items, Editiereinheit und die Datenhaltung.

Die Systemkontrolle und -steuerung ist unabhängig von der Darstellung der Inhalte. Sie greift direkt auf die Datenhaltung und auf die Funktion der graphischen Oberfläche zu. Ein Item erzeugt graphische Ausgaben und nimmt Benutzereingaben entgegen. Jedem Item wird dazu ein Fenster auf der graphischen Oberfläche zugeordnet. Die Editiereinheit wird zur Manipulation der Inhalte in der Datenbank genutzt.

Das Leistungsspektrum einer Editiereinheit reicht von der Änderung und dem Austausch von Inhalten, bis hin zur vollständigen Erstellung eines Kiosk mit Ablaufstruktur und Layouts. Die Datenhaltungsschnittstelle stellt abstrakte Operationen (z.B. das Lesen und Setzen von Attributen) zur Verfügung. Die spezifischen Erweiterungen umfassen benötigte Peripheriegeräte und Kommunikationsmöglichkeiten, Online Dienste zum Aktualisieren von Kiosk-Inhalten, sowie zur Anpaßung der Benutzungsoberfläche an die speziellen Betreiberwünsche. Ein Kiosksystem ist meistens in einem Gehäuse untergebracht, das eine bedienungsfreundliche Anwendung durch das Netz erlaubt. [DGBK95]

Die Architektur eines Kiosksystems läßt sich in drei funktionale Schichten aufteilen. Die erste Schicht im System bildet die graphische Oberfläche, die zweite Schicht bildet die Implementierung der Objekte des Systems, die unterste Schicht beschreibt die Datenhaltung (Bild13.1).

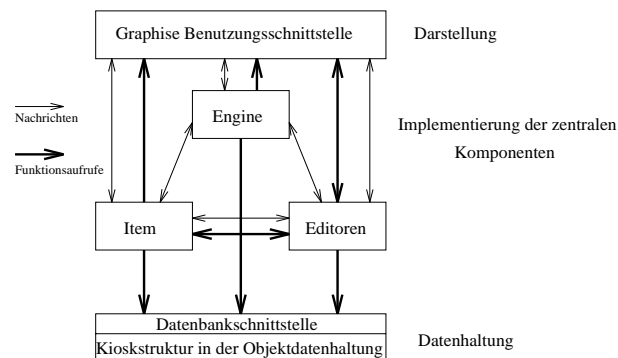


Abbildung 13.1: Architektur eines Kiosksystems

13.4 Anwendungsbereiche

13.4.1 Einsatzgebiete

Kiosksysteme können für unterschiedlichste Aufgabenstellungen eingesetzt werden. Es sollen im folgenden einige typische Anwendungsbereiche erläutert werden.

13.4.1.1 Bahnhöfe und Flughäfen

Kiosksysteme werden in Bahnhöfen oder Flughäfen als Auskunft- und Transaktionssysteme eingesetzt. Man kann über sie Auskünfte wie z.B. : An- und Abfahrtszeiten, Lageplan der Bahn- oder Flugsteige, Anschlußverbindungen, oder auch Informationen über Sonderfahrten oder -flüge erhalten. Der Benutzer kann unterschiedliche Fragen oder Transaktionen an das Kiosksystem stellen wie: An welchem Bahnsteig kommt der Zug aus Berlin an? Wo befindet sich der Schalter einer bestimmten Fluggesellschaft?

13.4.1.2 Messen, Ausstellungen, Museen

Man sucht immer neue Wege, Besucher zu informieren und zu leiten. Kiosksysteme werden in diesem Bereich als Informationssysteme eingesetzt und beinhalten Kataloge, die Auskünfte über verschiedene Punkte liefern, z.B. ein Plan der Messe, Händlerverzeichnis, Produktverzeichnis, Ausstellungsobjekte. Mittels eines Sachgebietskatalogs lassen sich darüber hinaus gezielt Informationen über einzelne Anbieter abrufen und ermöglicht so dem Besucher eine leichte Orientierung. Man kann an das System Fragen stellen, wie zum Beispiel: Wo befindet sich der Aussteller IBM? Welche Aussteller bietet das Produkt x an?

Ein geeigneter Platz für Kiosksysteme sind Museen. Man kann Informationen über die im Museum

vorhandenen Exponate mit Bildern, Videos und Hintergrundinformationen darstellen. Weiterhin kann ein solches System über Sonderveranstaltungen informieren. Es ist auch möglich zu fragen, welche Bilder von Pablo Picasso im Museum zu sehen sind, in welchen Räumen sie ausgestellt sind und wie man dort hin gelangt.

13.4.1.3 Einzelhandel

Im Einzelhandel sind Kiosksysteme noch nicht sehr verbreitet, obwohl hier vielfältige Anwendungsmöglichkeiten gegeben wären. Denkbar ist der Einsatz von Kiosksystemen als Point-of-Sale Kundeninformationssysteme in Warenhäusern oder Großmärkten. Mögliche Anwendungen sind: multimediale Produktpräsentationen oder Hinweise auf Sonderangebote, die den Kunde anlocken. Man kann auch den Lageplan einzelner Produkte vorstellen. Das erleichtert das Einkaufen für viele Kunden.

13.4.1.4 Banken

In Banken haben Kiosksysteme momentan die weiteste Verbreitung in Form von Bankautomaten gefunden (engl. Automated Teller Machine ATM). Ihre Funktionalität ist bisher größtenteils auf das Abheben und Überweisen eines bestimmten Geldbetrags oder das Ausdrucken des aktuellen Kontoauszuges beschränkt. Jedoch ist das Potential für den Einsatz von Kiosksystemen als multifunktionales Kundeninformations- und Transaktionssystem bei weitem noch nicht ausgeschöpft. Man kann sich vorstellen, daß später über künftige Systeme eine Anlagerberatung oder Kreditberatung abgewickelt werden könnte. Bei Bedarf könnte eine sogenannte Beratungsfunktion integriert werden, über die eine audiovisuelle Verbindung zu einem Kundenberater aufgebaut werden kann.

13.4.1.5 Immobilien

Für Immobilienmakler eignen sich Kiosksysteme als Informationssystem über aktuell angebotene Objekte mit z.B. Lagebeschreibung und Preisinformationen.

Alle diese Beispiele deuten an wie breit das Spektrum für Kiosksysteme heute bereits ist: Präsentation, Information, Beratung, Transaktion, Unterhaltung und Schulung.

13.5 Multimediale Kiosksysteme

13.5.1 Was ist Multimedia

Multimedia kennzeichnet die rechnergesteuerte, integrierte Erzeugung, Manipulation, Darstellung, Speicherung und Kommunikation unabhängiger Informationen mehrerer zeitabhängiger und zeitunabhängiger Medien. Es werden folgende Arten von Medien unterschieden: Perzeptionsmedien, Repräsentationsmedien, Präsentationmedien, Speichermedien, Übertragungsmedien, Informationsaustauschmedien. [Hol95]

13.5.2 Verteilte multimediale Kiosksysteme

13.5.2.1 Multimedia Kiosksystem

Im kleinen Lexikon der Informatik findet man für Multimedia folgende Erklärung:

“Sammlung von verschiedenen Datentypen, Beziehungen zwischen verschiedenen Datentypen, wie Synchronisation, Kombination und Konvertierung von einem in das andere Medium.” [Kle95]

Hier ist unter 'Multimedia' eine Vielzahl von Datentypen zu verstehen, die untereinander in einer Beziehung stehen. Sie müssen synchronisierbar, kombinierbar und konvertierbar sein. Unter Synchronität ist bei. das korrekte zeitliche Einsetzen eines Musikstückes zu einem Video zu verstehen. Eine Synchronisierung kann aber auch räumlich oder inhaltlich erfolgen. Die oben angegebenen Datentypen werden gewöhnlich als Repräsentationsmedien bezeichnet, da sie die rechnerinterne Darstellung eines Mediums repräsentieren.

Der Einsatz multimedialer Elemente im Kiosksystem hilft nicht nur, ein System benutzerfreundlich und attraktiv zu gestalten, oder durch Audio- und Videoclips besser auf sich aufmerksam zu machen, sondern es eröffnet auch neue Anwendungsbereiche. Komplizierte Sachverhalte können in Form von Kurzfilmen wesentlich einfacher und besser dargestellt werden. Informationen, die nicht nur gelesen sondern auch gehört und in einem Video gesehen werden, werden besser von den Konsumenten behalten. Interaktivität in Verbindung mit Multimedia kann diesen Effekt zusätzlich verstärken. Hat man etwas 'selbst gemacht' und sei es nur durch Klicken am Bildschirm, wird es noch einmal besser behalten, als wenn man die Information passiv vermittelt bekommt.

13.5.2.2 Verteilte Kiosksystem

Verteilte Kiosksysteme bieten gegenüber lokalen Systemen eine Reihe von Vorteilen. Daten können global gehalten und gepflegt werden und sind damit aktuell und auf allen Systemen einheitlich. Kiosksysteme können über Rechnernetze hinweg ferngewartet werden. Dies erspart Wege und Zeit im Vergleich zu der Alternative, die Systeme vor Ort warten zu müssen. Die Anbindung an andere Online Dienste ermöglichtes, Informationen anzubieten, bei denen es nahezu unmöglich wäre, sie aktuell zu halten, wenn sie lokal gespeichert wären.

13.5.2.3 Verteilte Multimediale Systeme

Die Kombination von Multimedia und Verteilung ermöglicht weitere interessante Alternativen und Vorteile. Interaktive Audio und Video Konferenzsysteme können eingebunden werden und erlauben dem Benutzer Online Kontakt (z.B. zu einer Beraterin) aufbauen zu können. Über 'Video on Demand Server' können aktuelle Videos und Filme auf ein Kiosksystem geladen und dort abgespielt werden. Über Videoeinspielungen entfernter Kameras können dem Benutzern aktuelle Informationen und Bilder vermittelt werden.

13.5.3 WWW als Kiosksystem

Die Architektur des World Wide Web ist sehr gut zur Implementierung verteilter multimedialer Kiosksysteme geeignet. Um zu verdeutlichen, wie die im WWW verwendete Architektur eingesetzt werden kann, um verteilte multimediale Kiosksysteme zu implementieren, werden einige Beispiele existierender WWW- basierter Kiosksysteme vorgestellt. **Beispiele für WWW Kiosksysteme**

- Universität Mannheim: Im Herbst 1993 wurde am Dekanat für Betriebswirtschaftslehre der Universität Mannheim ein Projekt mit dem Namen 'Informationssystem zur Verkürzung der Studienzeiten' ins Leben gerufen. In einer empirischen Analyse stellte sich heraus, daß fehlende Informationen der wichtigste Faktor für lange Studienzeiten sind.

Nach einer Befragung der Lehrstühle, nach deren Wünschen über Inhalte des Informationssystems, Anforderungen an ein integriertes Verwaltungssystem, sowie die verwendete Hard- und Software, wurde Anfang 1994 damit begonnen, ein Informationssystem auf Basis des World-Wide-Web zu entwickeln. Das Projekt wurde im Oktober 1994 auf der zweiten internationalen WWW-Konferenz in Chicago erstmals der Öffentlichkeit vorgestellt. Im Januar 1995 wurde die erste WWW-basierte Kiosksäule an der Universität Mannheim eingeweiht.

- Der Spiegel: Als einer der ersten Nachrichtenmagazine weltweit und als Vorreiter in Deutschland bietet der Spiegel Auszüge aus seinem Magazin online im World-Wide-Web an. Neben dem Titelblatt, der Titelgeschichte und dem kompletten Inhaltsverzeichnis werden außerdem einige ausgewählte Artikel im Netz bereitgestellt.
- 'Das Internet Shopping Center' zeigt ganz deutlich, wie das World-Wide-Web als mächtiges Verkaufsinstrument eingesetzt werden kann. Das gleiche Medium dient sowohl für verteilte Kiosksysteme, als auch für Online-Dienste über die Millionen von Internet Benutzern weltweit im Internet einkaufen können. Der Nachteil des Fernsehens als Verkaufsmedium ist, daß es nur unidirektional ist und die Kunden ein weiteres Medium, in diesem Fall das Telefon, verwenden müssen, um

einkaufen zu können. Das Internet, und speziell das WWW, bietet aber die Möglichkeit, interaktiv Transaktionen zu tätigen und über das gleiche Medium individuell Angebote einholen sowie bestellen zu können.

13.6 Bewertung

13.6.1 Vorteile eines Kiosksystem

13.6.1.1 Aus der Sicht der Dienstanbieter

Bevor ein Dienstanbieter sich dazu entschließt, ein Kiosksystem einzusetzen, wird er sich die Frage stellen: Lohnt sich die Investition? Welchen Nutzen bietet der Einsatz eines Kiosksystem? Welche Kosten entstehen?

- Im Verkauf- und Servicebereich sind ein Großteil, der von Kunden gestellten Fragen oder der gewünschten Informationen Routinefragen bzw. Standardinformationen. Kiosksysteme können diese Aufgabe übernehmen und damit zu einer Einsparung von Personalkosten beitragen. Oftmals kann aber auch qualifiziertes Personal Fragen nicht unmittelbar und ohne das Kiosksystem auch nicht beantworten. Kiosksysteme können so ausgelegt werden, daß sie Zugriff auf eine Datenbank haben, die dem Kunden ein breites Spektrum an aktuellen Informationen zur Verfügung stellt.
- Banken, Versicherungen oder auch öffentliche Einrichtungen haben häufig Öffnungszeiten, die es vielen Leute schwer machen, den Service in Anspruch zu nehmen. Informations- bzw. Serviceverfügbarkeit durch ein Kiosksystem, das rund um die Uhr steht, kann hier Abhilfe schaffen.
- Ein weiterer Vorteil ist, daß Kiosksysteme an Plätzen und Orten eingesetzt werden können, an denen anderweitig in dieser Form kaum oder nur sehr schwer Dienstleistungen oder Informationen angeboten werden können. Geringer Platzbedarf bei hohem elektronischem Informations-, Waren- oder Dienstleistungsangebot, sowie die Tatsache, daß kein Verkaufs- bzw. Servicepersonal eingesetzt werden muß, machen dies zusätzlich attraktiv.
- Die Integration neuer Medien wie Audio und Video ermöglicht Kioskbetreibern, ihre Informationen, Waren oder Dienstleistungen attraktiver und verständlicher zu repräsentieren. Viele Menschen sind außerdem mit dem Umgang dieser Medien durch Fernsehen und Rundfunk vertraut und reagieren darauf positiv.
- Mehrsprachige Varianten eines Kiosksystem sind relativ einfach zu realisieren. Dadurch wird nicht nur ein größerer Kundenkreis angesprochen, sondern teilweise wird es auch Personen ermöglicht, an Informationen zu kommen, die sie sonst kaum oder nur schwer bekommen hätten.
- Im Gegensatz zum persönlichen Berater, der teilweise Subjektivität vermittelt, wirkt die Information, die über ein Kiosksystem angeboten wird, überwiegend neutral und objektiv.

Wichtig für ein Kiosksystem ist es, daß über eingebaute Statistikfunktionen ein Feedback über das Benutzerverhalten möglich ist. Durch Statistik über Anzahl der Zugriffe und die jeweilige Verweildauer können z.B. besonders interessante Seiten im System ausgemacht werden, was wertvolle Rückschlüsse über das Kunden- und Konsumentenverhalten liefern kann. Aber auch unattraktive Seiten können darüber identifiziert und eventuell überarbeitet werden.

13.6.1.2 Aus der Sicht der Dienstbenutzer

Nicht nur Dienstanbieter, sondern auch Dienstbenutzer, also Kunden, Konsumenten oder Informationssuchende, können aus dem Einsatz von Kiosksystemen Nutzen und Vorteile ziehen. Neben der ständigen Verfügbarkeit sind hier vor allem Aktualität, Anonymität, die Möglichkeit der interaktiven Gestaltung des Informationsprozesses sowie Zugriffsgeschwindigkeit auf die gewünschte Information bzw. den gewünschten Service die ausschlagenden Faktoren.

- Durch gezielte Beratung und Informationsbeschaffung in einem Kiosk vor Ort kann der Kunde beim Einkauf in einem Kaufhaus z.B. bereits 'online' Produkte vergleichen und aussuchen, wobei anschließend Wege und Zeit eingespart werden können. Das System kann konkret Informationen über Preis und Verfügbarkeit von Produkten geben oder anhand bestimmter Kriterien die günstigsten Angebote aus dem Sortiment vorschlagen. Eine anschließende Wegebeschreibung, wo die Produkte zu finden sind, erleichtert das Einkaufen zusätzlich.
- Bankgeschäfte sind häufig Routineaufgaben, die ohne persönliche Beratung an einem Kiosksystem schneller abzuwickeln sind als am Bankschalter.
- In Öffentlichen Nahverkehr ist es schon lange üblich, Fahrkarten am Fahrkartenautomaten zu lösen. Auch diese Automaten werden immer weiter ausgebaut, und die Fahrgäste können teilweise bereits zusätzlich zum Fahrkartenverkauf Informationen über Fahrpläne, Streckennetz, Anschlußverbindungen etc. abrufen.
- Beim Lösen einer Eintrittskarte für eine Theaterveranstaltung, ein Konzert oder eine Sportveranstaltung ist es interessant zu wissen, welche Sicht von dem einen oder anderem Platz aus vorhanden ist. Hier können interaktive Verkaufskioske durch eine Videoeinspielung Beispiel geben, anhand derer der Kunde besser entscheiden kann (in den meisten Fällen entscheidet sich der Kunde für teurere Karten).

13.6.1.3 Vorteile spezieller Kiosksysteme

Neben den oben genannten, allgemeinen, von Art und Einsatzgebiet der Kioske weitgehend unabhängigen Vorteilen, die Dienstanbieter durch den Einsatz von Kiosksysteme haben, gibt es für spezielle Kiosksysteme weitere Vorteile, die im folgenden an zwei konkreten Beispielen dargestellt werden sollen.

1. Verkaufskioske in Verbrauchermärkten oder Warenhäusern:

- Durch ansprechende Produktpräsentation können beim Kunden oder Konsumenten vor Ort gezielt Bedürfnisse geweckt werden.
- Durch multimediale Produktpräsentation können Produkte angeboten werden, die aufgrund der benötigten Stellfläche sonst nicht immer verfügbar und nicht angeboten werden (z.B. große Möbel, Teppiche).
- Interaktive Verkaufskioske geben dem Kunden die Möglichkeit, Produkte individuell zusammenzustellen und sich ein Bild vom Ergebnis machen zu können (z.B. Kücheninstallation, Wohnungseinrichtungen).

1. Kioske in Bankenbereich

- Kioske ermöglichen diskrete und schnelle Abwicklung vieler Bankgeschäfte.
- Handschriftliche Überweisung und Dauerraufträge werden durch die direkte Eingabe am Kiosk ersetzt. Damit kann nicht nur mindestens ein Arbeitsschritt sowie Zeit eingespart, sondern auch die Fehleranfälligkeit deutlich reduziert werden.

13.6.2 Nachteile eines Kiosksystem

- Routinearbeiten können auf Kiosksystemen durchgeführt werden, wodurch Mitarbeiter entlassen werden.
- Kunden können sich über Produkte und Dienstleistungen informieren und so mit gezielten Fragen in ein beratendes Gespräch gehen oder dieses ganz vermeiden.
- Die Akzeptanz eines Kiosksystem ist noch sehr niedrig.
- Ein solches System kostet noch sehr viel Geld und seine Wartung benötigt spezialisierte Kräfte, die auch teuer sind.

13.6.3 Bewertung des Fraunhofer-Instituts

Ziel der Studie war es, relevante Kriterien für den Einsatz von multimedialen Kioskterminals zu ermitteln. Die befragten Experten waren überwiegend der Meinung, daß die heutigen Kiosksysteme 'keinen Spaß machen', 'immer das Gleiche' sowie 'anstrengend', 'langweilig', 'monoton' und 'enttäuschend' seien. Zur Verbesserung der Anwendungen müßten mehr Zeit und Aufwand in die Entwicklung investiert sowie längere Testphasen durchgeführt werden. [POS97]

13.7 Literatur

- [DGBK95] Dr. Ralf Steinmetz Dr. Gerold Blakowski und Daniel Koehler. Architektur und Implementierung eines verteilten Multimedia Kiosksystems. *it+ti*, 4:35–43, aug 1995.
- [Hol95] Wieland Holfelder. *Multimediale Kiosksysteme*. Vieweg, Braunschweig/Wiesbaden, 1995.
- [Kle95] *Kleines Lexikon der Informatik*. Oldenburg-Verlag, 1995.
- [Mey75] *Meyers enzyklopaedisches Lexikon*. 1975.
- [NS96] Joachim Niemeier und Martina Schaefer. Kiosk Systeme - Point of Information. In *HMD: Theorie und Praxis der Wirtschaftsinformatik: Multimedia Anwendungen*, number 188, Seiten 72–87. 1996.
- [POS97] *POS Point of Sale*. URL: <http://www.hightext.cube.net/pos.htm>, 4.2.1997.