

- [Piegl 95] Piegl, L.; Tiller, W.: The NURBS book. Berlin: Springer 1995
- [Porter 84] Porter, T.; Duff, T.: Compositing digital images. SIGGRAPH '84 Proceedings. ACM Computer Graphics 18/4 (1984) 253-259
- [Prusinkiewicz 96] Prusinkiewicz, P.; Lindenmayer, A.: The algorithmic beauty of plants. New York, NY: Springer 1996
- [Requicha 85] Requicha, A. A. G.; Voelcker, H. B.: Boolean operations in solid modeling: boundary evaluation and merging algorithms. Proceedings of the IEEE 73/1 (1985) 30-44
- [Rohlf 94] Rohlf, J.; Helman, J.: IRIS performer: a high performance multiprocessing toolkit for real-time 3D graphics. SIGGRAPH '94 Proceedings. ACM Computer Graphics 28 (1994) 381-395
- [Samet 90] Samet, H.: The design and analysis of spatial data structures. Reading, Mass.: Addison-Wesley 1990
- [Shoemake 85] Shoemake, K.: Animating rotation with quaternion curves. SIGGRAPH '85 Proceedings. ACM Computer Graphics 19/3 (1985) 245-254
- [Sillion 95] Sillion, F.; Drettakis, G.; Soler, C.: A clustering algorithm for radiance calculation in general environments. In: Hanrahan, P.; Purgathofer, W. (eds.): Rendering techniques '95. Wien: Springer 1995, 196-205
- [Sowizral 98] Sowizral, H.; Rushforth, K.; Deering, M.: The java 3D API specification. Amsterdam: Addison-Wesley Longman 1998
- [Strothotte 02] Strothotte, T.; Schlechtweg, S.: Non-Photorealistic computer graphics: modeling, rendering, and animation. San Francisco: Morgan Kaufmann 2002
- [Ulichney 88] Ulichney, R.: Digital halftoning. Cambridge, Mass.: MIT Press 1988
- [Upstill 90] Upstill, S.: The RenderMan companion. Reading, Mass.: Addison-Wesley 1990
- [Veach 97] Veach, E.; Guibas, L. J.: Metropolis light transport. SIGGRAPH '97 Proceedings. ACM Computer Graphics 31 (1997) 67-76
- [Vince 93] Vince, J.: 3-D computer animation. Reading, Mass.: Addison-Wesley 1993
- [Vince 95] Vince, J.: Virtual reality systems. Wokingham: Addison-Wesley 1995
- [VRML 97] The virtual reality modeling language: International standard ISO/IEC 14772-1:1997
- [Wernecke 95] Wernecke, J.: The inventor toolmaker. Reading, Mass.: Addison-Wesley 1995
- [Wyvill 86] Wyvill, G.; McPheeters, C.; Wyvill, B.: Data structures for soft objects. The Visual Computer 2 (1986) 227-234
- [Young 89] Young, D. A.: X window systems - programming and applications with Xt. Englewood Cliffs, NJ: Prentice Hall 1989
- [Zyda 93] Zyda, M. J.; Pratt, D. R.; Falby, J. S.; Lombardo, C.; Kelleher, K. M.: The software required for the computer generation of virtual environments. Presence 2/2 (1993) 130-139

## 2 Multimedia

### M. Mühlhäuser

2.1	Charakteristika, Paradigmen	855
	Definition - Realzeit-Anforderungen - Grobklassifikation von Werkzeugen - Anwendungen	
2.2	Grundlegende Konzepte	857
	2.2.1 Synchronisation	857
	2.2.2 Dienstgüte	859
	2.2.3 Ströme	861
	2.2.4 Zeitkapseln	861
	2.2.5 Konfigurationsverwaltung	862
2.3	Handhabung von Multimediadaten	862
	2.3.1 Interaktion	862
	2.3.2 Codierung	863
	2.3.3 Entwicklung und Ausführung	866
	2.3.4 Speicherung	868
	2.3.5 Interpretation	868
2.4	Systemunterstützung	870
	2.4.1 Netzwerke und Kommunikationsdienste	870
	2.4.2 Betriebssysteme	871
2.5	Hypertext / Hypermedia	872
	2.5.1 Grundbegriffe	872
	2.5.2 Das Dexter-Modell	872
	2.5.3 Weitergehende Aspekte	873
	2.5.4 Das World-Wide Web als Hypertextsystem	874
	Allgemeine Literatur	876
	Spezielle Literatur	876

### 2.1 Charakteristika, Paradigmen

**Definition.** Nach einer Definition von Steinmetz ist ein Multimedia-System durch die *rechnergestützte koordinierte Handhabung mehrerer, zumindest teilweise zeitabhängiger Medien* gekennzeichnet. Zeitabhängige Medien sind solche, für deren Darstellung ein inhärenter Zeitbegriff wesentlich ist, z.B. Audio und Video. „Koordiniert“ bezieht sich auf zeitliche Zusammenhänge zwischen mehreren Medien (siehe Abschnitt 2.2.1).

**Realzeit-Anforderungen.** Multimediasysteme stellen in zweierlei Hinsicht meist schwächere Anforderungen als herkömmliche Realzeitsysteme: (1) der Ressourcenbedarf ist vorhersagbar, wenn mit konstanter Rate Bedienzeiten konstanten Umfangs angefordert werden, z.B. jede 25stel Sekunde die Bearbeitung eines Videobildes; (2) Überschreitungen von Fristen werden meist nicht als fehlerhaftes Verhalten interpretiert, sondern als graduelle Verschlechterung der Medienqualität. Medien mit den genannten Eigenschaften (konstante Bedienzeiten mit konstanter Rate ohne „harte“ Realzeitanforderungen) werden in der Multimedia-Literatur oft als *isochron* bezeichnet; das typische Beispiel sind zeitabhängige Medien mit konstantem Kompressionsfaktor. Computergenerierte Medien werden meist so modelliert, daß sich ein nicht-isochroner Datenstrom ergibt; Beispiele

sind Animationen, maschinenlesbare Aufzeichnungen interaktiver Bildschirmsitzungen und interaktive virtuelle Welten.

**Grobklassifikation von Werkzeugen.** Bei der Entwicklung von Multimedia-Software treffen die gestalterische Arbeit eines Autors oder Designers und die ingenieurmäßig oder algorithmisch geprägte Tätigkeit eines Software-Entwicklers aufeinander. Daher kann man bei der Entwicklungsunterstützung grob Autorenwerkzeuge (Autorensysteme oder Hypermediasysteme, das Arbeitsergebnis wird meist *Dokument* genannt) und Software-Entwicklungsumgebungen (Mikrowelten, Frameworks) unterscheiden.

**Autorensysteme.** Diese erlauben die Erstellung von Multimedia-Dokumenten, welche bei den Endbenutzern mehr oder weniger interaktiv „dargestellt“ werden. Meist wird hierzu eine der folgenden drei Metaphern verwendet.

- Bei der *Zeitleisten*-Metapher werden ähnlich einem komfortablen Videoschnittsystem Medienfragmente (Videoclips, Audiotracks) miteinander quasi zu einem fortlaufenden Film kombiniert. Interaktivität und eingabeabhängige Sprünge sind meist möglich, aber konzeptionell dominiert ein Handlungsstrang.
- Bei der *Karten*-Metapher werden Präsentationen als Stapel von Karteikarten aufgefaßt, sequentielles Lesen und Sprünge zu entfernten Karten (z.B. mit Hilfe von programmierten Knöpfen) sind intuitiv erfaßbar. Zeitabhängige Medien sind nur als Teil einer Karte möglich, der synchrone Fortschritt mehrerer zeitabhängiger Medien ist schwieriger auszudrücken (wenn überhaupt) als bei der Zeitleisten-Metapher.
- Bei der *Flußdiagramm*-Metapher werden Elemente des Steuerflusses wie Videopräsentation, Abfrage oder Verzweigung zu einem Dokument verknüpft. Häufig werden Elementtypen graphisch durch anschauliche Sinnbilder unterschieden. Oft stehen zum Entwurf einzelner Elemente Unterwerkzeuge zur Verfügung, beispielsweise ein Videoreditor nach der Zeitleisten-Metapher. Die Flußdiagramm-Metapher wird häufig bei Autorenwerkzeugen für Lehrsoftware eingesetzt. Diese unterstützen Konzepte des computerunterstützten Unterrichts, erreichen aber meist nur implementierungsnahes Niveau, nicht die in der Softwaretechnik übliche Analyse- und Entwurfsunterstützung.

**Hypermediasysteme.** Diese Systeme werden in Abschnitt 2.5 ausführlicher besprochen. Sie erlauben die flexible Verknüpfung von Informationseinheiten (Dokumentfragmenten) mittels Verweisen und erweitern entweder die Karten-Metapher oder lehnen sich stärker an das Konzept des herkömmlichen Textdokumentes (Artikel, Buch) an.

**Mikrowelten.** Der Begriff bezeichnet „geschlossene“ Entwicklungsplattformen für Einzelrechner. Sie vereinen ein universelles Multimedia-Datenformat mit Werkzeugen und Laufzeitunterstützung gemäß einer umfassenden Architektur. Beispiele sind Apple Quicktime und die Multimedia-Erweiterungen von Microsoft zu Windows.

**Frameworks.** Für offene Entwicklungsplattformen für (vor allem verteilte) Multimedia-Anwendungen wird zunehmend der Begriff *Framework* verwendet. Entwicklungen kommen bisher vor allem aus dem akademischen Bereich, Standardisierungsbemühungen der International Multimedia Association IMA gelten als gescheitert, Sun's Java Media Framework JMF ist erfolgversprechender. Sie realisieren meist die Mehrzahl der im nächsten Abschnitt aufgezählten grundlegenden Konzepte; vor allem unterstützen sie die Kopplung vordefinierter Komponenten zu Anwendungen. Mikrowelten unterstützen zunehmend verteilte Systeme und Offenheit, Frameworks kommen nicht ohne spezifisches Medien- und Verarbeitungsmodell aus, so daß sich beide annähern.

**Anwendungen.** Anwendungsfelder betreffen praktisch das ganze Spektrum der Informatik. Neben der Multi-/Hypermedia-Dokumentbearbeitung spielen hauptsächlich verschiedene Varianten von sog. *Telemedia-Anwendungen* eine Vorreiterrolle:

- **Multimedia-Mitteilungssysteme** erweitern elektronische Mitteilungsdienste (electronic mail) um Unterstützung für zeitabhängige Medien. Kritisch ist hier unter anderem die Handhabung der großen Datenmengen (Zwischenspeicherung, Kopierenreduktion).
- **Schreibtisch-Videokonferenzen** werfen Probleme auf wie die geeignete Verteilung von Medienströmen an viele Teilnehmer (Multicast-Mechanismen, Komposition der darzustellenden Videobilder) und geeignete Begleitwerkzeuge. Für letztere haben sich folgende Begriffe eingebürgert: (1) *whiteboard*: Software für kooperatives Schreiben und Zeichnen in einem gemeinsamen Dokument; (2) *application-sharing*: Darstellung eines Fenstersystem-basierten Programmes bei vielen Teilnehmern im Multiplex und zur Koordination der Teilnehmer-Eingaben; (3) *Konferenzmanagement*; (4) *kooperationsbewußte Anwendungen*: Spezialsoftware für kooperatives Arbeiten.
- **Digitales Fernsehen** im Sinne sog. *Video-on-Demand*-Verteilungssysteme wird dann besonders schwierig, wenn Teilnehmer (Haushalte) ganz individuell Videos abrufen können (siehe Abschnitt 2.3.4). „Near-Video-on-Demand“ bündelt solche Zugriffe, erlaubt Abruf aber nur in Zeitraster und macht Videorecorder-ähnliche Bedienung (Anhalten, Zurückspulen,...) unmöglich, da pro Zeitraster mehrere Haushalte bedient werden. Das vieldiskutierte *interaktive* Fernsehen ist bislang auf wenige Anwendungsfelder (home banking, teleshopping, Spiele, Lehrsoftware) bzw. eher marginale Erweiterungen des Paradigmas Fernsehen (Wahl unter mehreren Kamerapositionen, komfortablere Zuschauer-Meinungsumfragen) beschränkt geblieben. Neuere Standards können den Weg zur Verschmelzung von interaktiver sog. Edutainment-Software und digital ausgestrahltem herkömmlichem Fernsehen ebnen, siehe Abschnitt 2.3.2.
- **Kiosk-Systeme** kombinieren Multi- bzw. Hypermediadokumente und einfache Benutzerschnittstelle; sie wachsen zunehmend mit den obengenannten Anwendungsfeldern zusammen.

## 2.2 Grundlegende Konzepte

Es gibt wenig Konsens darüber, welche Fähigkeiten ein multimediafähiges System ausmachen. Nachfolgend werden fünf Kernbereiche unterschieden, für die ein Multimedia-Unterstützungssystem Konzepte und Modelle an der Schnittstelle zur Anwendung anbieten sollte: *Synchronisation*, *Dienstgüte*, *Ströme*, *Zeitkapseln* und *Konfigurationsverwaltung*. „Unterstützungssystem“ steht dabei für ein multimediafähiges Betriebssystem oder Netz oder ein Werkzeug im obengenannten Sinne [Gibbs 95, IMA 95].

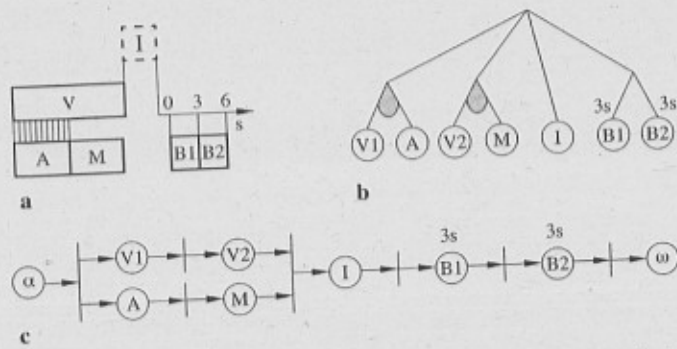
### 2.2.1 Synchronisation

Die Literatur versteht unter (*Medien*-)Synchronisation die zeitliche, inhaltliche und räumliche Abstimmung mehrerer Medien in einer Multimedia-Darstellung; die räumliche Anordnung beschränkt sich natürlich auf visuelle Medien. Inhaltliche Synchronisation bezieht sich auf mehrere zeitgleiche Präsentationen von Anwendungsdaten; dort müssen bei benutzergesteuerten oder systeminternen Veränderungen alle betroffenen Präsentationen aktualisiert werden (siehe dazu auch das in Kapitel D12.1) beschriebene MVC-Schema). Neben der sog. *Interobjektsynchronisation* mehrerer Medien muß auch die sog. *Intraobjektsynchronisation* der Verarbeitungseinheiten eines einzelnen zeitabhängigen Mediums gewährleistet werden.

Bisweilen ist die zeitliche Anordnung mehrerer Medien implizit durch die Herkunft gegeben, z.B. bei gleichzeitig aufgenommenen Audio- und Videodaten (*Live-Synchronisa-*

tion). Wenn nicht, dann muß sie vom Entwickler spezifiziert werden (*synthetische Synchronisation*). Zeitbezogene Aspekte von Synchronisationsspezifikationen sind problematischer als räumliche, inhaltliche Synchronisation wirkt kaum Multimedia-spezifische Fragen auf; dieses Kapitel beschränkt sich daher auf zeitliche Aspekte. Dabei können isochrome Medienströme gemischt werden mit interaktiven Objekten, zeitunabhängigen Objekten und nicht-isochronen Medien.

Zu den Anforderungen an Methoden zur Synchronisationsspezifikation gehören Intuitivität, Erhaltung der logischen Einheit von Medienobjekten, Hierarchiebildung (Einkapselung von Teilspezifikationen) und Ausdrucksstärke. Weiter sollten Details der Synchronisation festgelegt werden können wie Kohärenz (lippensynchrones Audio-Video ist genauer zu synchronisieren als Video mit Hintergrundmusik) oder verteilungsrelevante Parameter (Zusammenführung von Medien bei Quelle oder Ziel, Ausnahmebehandlung bei Leistungseinbruch). Wichtige Methoden werden nachfolgend kurz beschrieben (siehe Bild 1).



**Bild 1** Synchronisationsspezifikation: a Referenzpunkt-Methode, b hierarchisch, c Petrinetz. Parallel zu einem Video V läuft zunächst ein lippensynchrones Audio A, dann Hintergrundmusik M. Dann wird am Schaltknopf I eine Benutzereingabe erwartet, bevor zwei Bilder B1 und B2 je 3 s lang gezeigt werden. V1 und V2 sind zwei Hälften von V.

**Hierarchische Spezifikation.** Diese macht sich zu eigen, daß im Kern nur zeitlich parallele und zeitlich sequentielle Anordnung von Medien möglich ist. Entsprechend werden dort Medien als Blätter eines Baumes angeordnet. Ein gemeinsamer Vater-Knoten gibt an, daß alle darunterliegenden Medien (Blätter) bzw. Gruppen von Medien (Zwischenknoten) gleichzeitig bzw. alle zeitlich nacheinander dargestellt werden; dazu werden zwei Arten von Unterbäumen unterschieden. Ein Hauptnachteil dieser Methode liegt darin, daß Medienobjekte bei nur teilweiser zeitlicher Überlappung unnatürlich zerteilt werden müssen.

**Balkenorientierte Methoden.** Bei allen drei Varianten dieser Methode werden zeitabhängige Medien graphisch durch einen Balken „von links nach rechts“ veranschaulicht. Im einfachsten Fall beziehen sich alle Balken auf eine gemeinsame reale Zeitachse (*Zeitachsen-Methode*), im weitest verbreiteten Fall auf ein mehrdimensionales Koordinatensystem (*Referenzachsen-Methode*). Dessen Achsen müssen sich (wie Taktstriche bei Noten) nicht immer auf zeitlich äquidistante Maßeinheiten beziehen und erlauben im Gegensatz zur Zeitachsen-Methode interaktive Objekte unbekannter Dauer. Die *Referenzpunkt-Methode* schließlich kommt weitgehend ohne Koordinatensystem aus, da Medien relativ zu

einander angeordnet werden. Soweit zeitabhängige Medien in die Spezifikation involviert sind, geben sie den zeitlichen Ablauf vor. Nur wo jene fehlen (Beispiel: reine Dia-Show), werden Zeitgeber-Objekte als Zeitachsen benutzt. Diese Methode erfüllt alle genannten Anforderungen weitgehend.

**Ereignistabellen.** In einigen Systemen und Standards wurden die bekannten Ereignistabellen eingesetzt, wobei in eine Matrix aus Medien und auslösenden Ereignissen (wie interaktiven Eingaben, Beginn oder Ende einer Mediendarstellung, Zeitgeber-Ereignissen) Folgeereignisse eingetragen werden; letztere umfassen u.a. Starten eines Zeitgebers oder wiederum Beginn oder Ende einer Mediendarstellung. Dieser Ansatz hat sich bei komplexeren Problemen als schlecht intuitiv erfaßbar, ausdruckschwach und wenig kompakt erwiesen. Als kompaktere Variante wurden *Ereignislisten* vorgeschlagen.

**Skriptsprachen.** Speziell entwickelte Skriptsprachen zeichnen sich naturgemäß durch höheren Lernaufwand, aber auch besondere Ausdruckskraft aus. Insbesondere können Standard-Operationen auf den Medien in den Sprachumfang einbezogen werden. So läßt sich aus Melodiefragmenten per Skript ein vollständiger Kanon durch Synchronisations-, Iterations- und musikspezifische Operationen (wie Transposition) erstellen.

**Formale(re) Methoden.** Petrinetz-Erweiterungen wurden vielfach zur Synchronisationsspezifikation vorgeschlagen oder entwickelt. Bei nichttrivialen Aufgaben werden Petrinetze schnell unübersichtlich, ihr Wert als anwendernahe Beschreibungsmethode ist daher umstritten. Auf Grund der Möglichkeiten zur formalen Prüfung und Umsetzung kommen sie aber in Frage als Zwischenstufe zwischen einer Spezifikation mit einer anderen Methode und deren Umsetzung in Ablaufpläne des Laufzeitsystems.

**Intervallbasierte Methoden.** Betrachtet man die Präsentation eines Mediums als Zeitintervall, so können zwei solche Intervalle A und B in verschiedenen Beziehungen zueinander stehen. Eine einfache Betrachtung unterscheidet sechs Möglichkeiten: A liegt zeitlich vollständig vor B, B folgt unmittelbar auf A, B folgt A überlappend, B läuft vollständig während A ab, A startet B und A beendet B. Ausdrucksstärkere Modelle beziehen die Verzögerungen zwischen Anfang oder Ende von A und Anfang oder Ende von B mit ein und berücksichtigen eine oder mehrere solcher Verzögerungen in verschiedenen Beziehungstypen. Graphische und skriptsprachenbasierte Spezifikation wurden vorgeschlagen. Offensichtlich lassen sich andere genannte Methoden mit intervallbasierten kombinieren oder zur einfacheren Erstellung von Ablaufplänen dahin übersetzen.

**Beispiele.** Das bekannte Autorensystem Macromedia Director mischt in seiner Sprache Lingo Elemente einer Programmier- und einer Skriptsprache. Das „time coordinate system“ TCS des Systems Quicktime der Firma Apple entspricht der Zeitachsen-Methode. Im ISO-Standard MHEG für Multimedia- und Hypermedia-Präsentationen wird die Referenzachsen-Methode verwendet, zusätzlich können Ereignislisten angelegt werden; zur internen Synchronisation können Ereignisobjekte verschickt werden. Ansätze zur Synchronisation in Hypermedia-Systemen werden in Abschnitt 2.5 besprochen, das dort erwähnte SMIL folgt der hierarchischen Methode; der in Abschnitt 2.3.2 besprochene Standard MPEG-4 enthält eine intervallbasierte Spezifikationssprache.

## 2.2.2 Dienstgüte

**Bedeutung und Eigenschaften.** Dienstgüte (*Quality-of-Service, QoS*) bezeichnet quantifizierbare Eigenschaften eines Dienstes; die einzelnen Eigenschaften werden als Dienstgütemerkmale oder -parameter bezeichnet. Wichtig wird der Begriff dann, wenn der Dienstnutzer Dienstgütemerkmale mitbestimmen kann. Merkmale wie Durchsatz und Restfehlerrate werden seit Jahren für Kommunikationsdienste betrachtet, sind in konven-

tionellen Netzen aber nur beschränkt variabel. Multimedia-Dienstgütermerkmale wie Bildauflösung oder Pixelfehlerrate können dagegen auf Grund der Toleranz und Subjektivität der menschlichen Wahrnehmung stark variiert werden. Die hohen Datenvolumina machen zudem ein Abwägen zwischen Güte und Kosten interessant. Drei Charakteristika von Dienstgütermerkmalen seien erwähnt:

- **Stochastik:** Dienstgütermerkmale sind meist statistische Größen, von denen z.B. Maxima, Mittelwerte oder andere Verteilungskenngrößen interessieren.
- **Konkurrenz:** Offensichtlich konkurrieren Dienstgütermerkmale untereinander und mit den abstrakten Kosten im Sinne einer Optimierungsfunktion. Langfristig wird an leistungsfähigen Unterstützungssystemen gearbeitet, die die dynamische Optimierung der Dienstgüte voll automatisieren („QoS-Broker“). Bislang werden meist zwischen Anwendung und Unterstützungssystem in einem iterativen Prozeß Dienstgütermerkmale *ausgehandelt* („QoS-Negotiation“). Bei der einfachsten Verhandlungsform gibt die Anwendung einen gewünschten Satz von Dienstgütermerkmalen vor, das Unterstützungssystem ermittelt dann gemäß einem Protokoll unter seinen Komponenten die (möglicherweise niedrigere) Dienstgüte, die es tatsächlich anbieten kann, und die Anwendung hat nur noch das Recht, dieses Angebot anzunehmen oder auf die Dienstleistung zu verzichten.
- **Schichtung:** Die subjektive Güte eines Videos auf Benutzerschicht, z.B. „bildschirmfüllendes, ruckfreies, lippensynchrones Video“ ist eine Funktion von objektiven Dienstgütermerkmalen auf der Schicht der Datenrepräsentation wie Auflösung, Farbtiefe und Bildwiederholrate sowie Codierungs-Spezifika wie Quantisierung, Abfolge von I-, P- und B-Frames usw. (siehe Kapitel B4.3.4). Diese wiederum sind eine Funktion von Dienstgütermerkmalen in den Schichten der Kommunikationsdienste und des Betriebssystems einschließlich der Peripheriegeräte.

**Merkmalklassen.** Qualitative Dienstgütermerkmale beschreiben das Vorhandensein oder fehlen von Fähigkeiten wie Multicast (Gruppensenden). Quantitative Merkmale werden durch numerische stochastische Parameter beschrieben. Einige der Dienstgütermerkmale sind seit langem bekannt. Andere wurden erst mit dem Aufkommen von Multimedia interessant, die Einteilung in folgende Klassen bleibt aber gültig:

- **Durchsatz** (auch: Datenrate, Bandbreite); für Vermittlungsrechner können Mittel- und Spitzenwerte der „Pakete pro Sekunde“ interessanter sein als „Bits pro Sekunde“
- **Verzögerung;** in diese Klasse fallen die beiden wohl wichtigsten Multimedia-spezifischen Merkmale:
  - **Jitter** (Zittern) ist definiert als zeitlicher Versatz zwischen dem Bearbeitungsende einer Verarbeitungseinheit und dem zugehörigen Nominalwert (z.B. bei Video mit 25 Hz Bildrate gegenüber demjenigen „Fünfundzwanzigstelsekunden-Tick“, der der Bildnummer entspricht).
  - **Skew** bezeichnet den Versatz synchronisierter Medienströme gegeneinander, der bei der Bearbeitung (Übertragung, Präsentation) entsteht. Die Summe der maximalen Jitter aller synchronisierten Ströme begrenzt den maximalen Skew.
- **Zuverlässigkeit;** hierzu zählen die *Restfehlerrate* als Wahrscheinlichkeit eines verälschten Bits trotz Fehlerkorrekturmechanismen und die *Zellenverlustrate* als Wahrscheinlichkeit, daß in einem zellenbasierten Netz eine ganze Zelle nicht ankommt.

Während die drei genannten Klassen allgemein gelten, können einzelne Merkmale komponentenspezifisch sein, wie Zykluszeit und Prozessorauslastung für Betriebssysteme. Das gilt auch für einige Multimedia-spezifische Parameter wie die maximale Positionierzeit zwischen einem Videobild und dem zugehörigen Audioblock (für Peripheriegeräte).

Dienstgütermerkmale, vor allem die eines aufzeichnenden oder komprimierenden Dienstes, beeinflussen unmittelbar Medienparameter wie Auflösung. Diese Parameter bezeichnet man zusammen auch als *Medienqualität*.

Als Beispiel abstrakter Kostenrelationen sei die Datenrate zweier typischer Audio-Dienstgüten verglichen. Digitale Telefonkanäle arbeiten mit einer Abtastrate von 8 kHz und quantisieren die Abtastwerte in 8-Bit-Werten, benötigen also 8 kHz·8 Bit = 64 KBit/s; digitale Audio-CDs verwenden 44.1 kHz Abtastrate, 16-Bit-Quantisierung und Stereo, also 1411.2 KBit/s Datenrate. Die Kostenrelation ist folglich 1411.2 : 64, also rund 22:1

### 2.2.3 Ströme

Die Datenübergabe zwischen Komponenten einer Multimedia-Anwendung unterscheidet sich von der herkömmlichen Parameterübergabe bzw. Datenübertragung, vor allem bei isochronen Medien. Für diese will man nicht jede einzelne Datenübergabe (z.B. für jedes Videobild) ausprogrammieren, sondern die Summe aller Datenübergaben in ihrem zeitlichen Ablauf spezifizieren. Zudem sind zur Gewährleistung isochronen Verkehrs umfangreiche Vorarbeiten wie Ressourcenreservierung notwendig, was verbindungsorientierte Datenübertragung nahelegt. Hierfür hat sich der Begriff *Strom* (stream) eingebürgert.

Derzeit werden verbindungslose Programmiersprachen und -paradigmen für verteilte Systeme bevorzugt, wie verteilte objektorientierte Programmierung oder Prozedurfernaufruf. Diese decken sich schlecht mit dem verbindungsorientierten Strom-Konzept: bekannte Multimedia-Erweiterungen für Prozedurfernaufruf führen z.B. Ströme ein, die durch je einen Prozedurfernaufruf für Verbindungsauf- und abbau gekapselt werden. In Zukunft könnten sich hier neue Programmierparadigmen durchsetzen.

Die Funktionalität von Strömen ist in verschiedenen Unterstützungssystemen unterschiedlich und betrifft u.a. folgende Aspekte: Strom-Verwaltung (Dienstgütwahl, Auf-/Abbau usw.), Manipulation (Geschwindigkeitswechsel, Zurückspulen usw.), Unterstützung von Gruppenkommunikation (multicast) und von Strömen über mehrere zwischengeschaltete Softwarekomponenten hinweg (multihop).

### 2.2.4 Zeitkapseln

Die Zeitabhängigkeit von Medien und Multimedia-Darstellungen schlägt auch auf die Anwendungsprogrammierung durch. Der Programmierer erhält die Möglichkeit, Zeitabhängigkeit für die Prozesse der Anwendung auszudrücken. Meist werden Leichtgewichtprozesse verwendet, z.B. aktive Objekte. Der dabei entstehende Prozeßbegriff soll als Zeitkapsel bezeichnet werden. Häufig wird Peripherie wie Kamera, Bildschirm, Kompressionshardware usw. homogen in das Zeitkapsel-Modell einbezogen.

Im einfachsten Fall werden Zeitgeber eingeführt, anwendungs- oder prozeßweit definiert. Fortgeschrittenere Konzepte erlauben sog. *synchrone Programmierung*, wozu Zeitkapseln in einer speziellen Programmiersprache als endliche Automaten spezifiziert werden; der Übersetzer kann dann für eine gegebene Zielplattform garantieren, daß bei gegebener konstanter Ereignisrate jede Ereignisbearbeitung vor der Ankunft des Folgeereignisses beendet ist. Entsprechende Erweiterungen der Objektorientierung wurden als „reaktive Objekte“ bezeichnet. Nachteil ist neben der erforderlichen speziellen Programmiersprache die Tatsache, daß nur Steuerprozesse als Zeitkapseln programmiert werden können. Dagegen wird die eigentliche Ereignisbearbeitung auf herkömmliche Weise erstellt und von den Steuerprozessen bei Zeitüberschreitung „beschleunigt“, indem z.B. die Darstellung der aktuellen Verarbeitungseinheit übersprungen wird. Trotz der vordergründig garantierten Bearbeitungszeit erhält der Programmierer so letztlich keinen Hin-

weis darüber, wie häufig solche Zeitüberschreitungen zur Laufzeit zu erwarten sind. Bekannte Modelle für Zeitkapseln sind daher noch entwicklungsbedürftig.

### 2.2.5 Konfigurationsverwaltung

Faßt man Multimedia-Anwendungen als Kombination von Programm-Komponenten (vor allem Zeitkapseln) und Strömen auf, dann benötigt der Programmierer eine Möglichkeit zum Aufbau von Anwendungen aus verteilten Komponenten. Dies wird z.B. in den in Abschnitt 2.1 erwähnten Frameworks unterstützt. Bei diesen läßt sich bemängeln, daß

- fast immer drei wesentliche Klassen von Komponenten unterschieden werden: für Produktion, Transformation und Konsum von Medien; jeder Ansatz „erfindet“ jedoch hierfür ein eigenes Modell;
- kaum ein Abstraktionsniveau angeboten wird, das die medien- und zum Teil hardware-abhängigen Aspekte der Komponenten zu verdecken gestattet; so müssen beispielsweise bei der Anwendungsentwicklung bereits Medienspezifika einer Kommunikationsbeziehung festgelegt werden, anstatt z.B. die Wahl zwischen Videokonferenz, reiner Audioverbindung und elektronischem Mitteilungsdienst zur Laufzeit zu ermöglichen;
- fast immer nur der Aufbau *einer* konkreten Konfiguration ermöglicht wird, anstatt z.B. die Anzahl der Komponenten eines bestimmten Typs erst zur Laufzeit festlegen zu müssen und dynamisch ändern zu können; die Softwaretechnik für verteilte Anwendungen hat hier vor Jahren schon einen wesentlich höheren Stand erreicht.

## 2.3 Handhabung von Multimediadaten

### 2.3.1 Interaktion

Multimedia-Interaktion erfordert, zumindest bis zur breiten Einführung digitaler Endgeräte, den Umgang mit analogen Signalen. Dabei werden Komponenten aus der Audio-Video-Technik verwendet, die sich von vergleichbarer Computerperipherie teilweise stark unterscheiden. Dies sei beispielhaft erläutert. Zwischen der Bildwiedergabe im Fernsehen und auf dem Computerbildschirm bestehen folgende Unterschiede: (1) Zeilenmodus: Beim Fernsehen wird jedes Bild in zwei größere Halbbilder zerlegt, die mit doppelter Bildwiederholrate dargestellt werden; so wird bei gleichbleibendem Bandbreitenbedarf das Flimmern deutlich reduziert; beim Computer werden Vollbilder benutzt, wodurch das Flimmern nicht flackern. (2) Bildwiederholrate: Beim Fernsehen werden 50 Halbbilder pro Sekunde dargestellt, beim Computer 70 oder mehr Vollbilder pro Sekunde. (3) Farbraum: Beim Fernsehen wird die YUV-Farbdarstellung, bei Computerbildschirm die RGB-Farbdarstellung benutzt (siehe Kapitel E1).

Auf die Vielfalt der sich schnell weiterentwickelnden Multimediaperipherie kann hier nicht eingegangen werden. Stattdessen sollen nur zwei Tendenzen und deren Folgen für Softwaretechnik und Anwendungsgebiete genannt werden:

- Vor allem für visuelle Medien wird weiterhin versucht, die „Realitätsnähe“ der Repräsentation zu verbessern, hinsichtlich Qualität (Farbtiefe, Auflösung usw.), Dimensionalität (2D → 3D) und Zeitabhängigkeit (Bild → Video).
- Die zunehmende Diversifikation interaktiver Komponenten, vom „Organizer“ im Westentaschenformat bis zur interaktiven Tafel, erhöht die Bedeutung *multimodaler Inter-*

*aktion*. Eine Modalität sei hier definiert als „Art und Weise der Interaktion“ welche sich auf die benutzten Sensor-Aktor-Kanäle bezieht (z.B. Audio, Video), auf die Signalrepräsentation (z.B. Sprache, Gestik) und die verwendeten Metaphern (z.B. Formulare, Menüs). Anforderungen an die Softwaretechnik betreffen umfassendere Interaktionsmodelle und den Wunsch, alternative Modalitäten zu unterstützen [Taylor 89].

Multimedia-Anwendungsdaten sind also von multimodaler *Anwendungskontrolle* (d.h. der Benutzeroberfläche) zu unterscheiden. Letztere wird auch auf Systemebene, also bei Fenstersystemen wie MS Windows oder X window, starke Veränderungen bringen.

### 2.3.2 Codierung

*Codecs*. Kapitel B4 diskutiert verlustfreie und verlustbehaftete Kompression und geht dabei auf Standards wie JPEG (für Festbilder), MPEG (für Video) inklusive MP3 (für Audio) und H.261/H.263 (für Videokonferenzen) ein. Dabei wird häufig das Kunstwort *Codec* benutzt, das von *Compressor/Decompressor* oder *Encoder/Decoder* herrühren kann. Es bezeichnet Kompressionsverfahren zusammen mit weiterreichenden Aspekten, die für den reibungslosen Austausch von komprimierten Multimediadaten in offenen Systemen wichtig sind. (In anderen Disziplinen wie der Nachrichtentechnik wird der Begriff in leicht abweichender Bedeutung verwendet). Solche weiterreichenden Aspekte seien nachfolgend beispielhaft diskutiert.

*Formate und Datenmengen*. Digitalisiert man ein analoges TV-Signal in Studio-Qualität verlustfrei einschließlich Austastlücken (*Dunkelphasen*), so ergibt sich nach Norm ITU-601 eine Datenrate von 216 MBit/s [Reimers 97]. Annähernd Videorecorder-Qualität erreicht das nur für Computer gedachte Videoformat CIF (*common interchange format*, ein ITU-Standard). Es hat 288·360 Punkte, 25 Hz Bildrate und 16 Bit pro Pixel, also immer noch eine Datenrate von mehr als 40 MBit/s, was bei einer Vierer-Videokonferenz die ATM-Übertragungsrate OC-3 (155 MBit/s) übersteigt. Heutige Videokonferenz-Systeme mit Hardwareunterstützung übertragen Bilder der Größe 144·180 Punkte (quarter CIF = QCIF) über einen ISDN-Kanal mit 64 KBit/s. Gegenüber CIF wie oben wird die Datenrate durch geringere Bildwiederholrate (10 statt 25 Hz) und 1/4-Auflösung auf rund ein Zehntel (also auf etwas über 4 MBit/s) reduziert, alles andere – also ungefähr der Faktor 65 – wird durch Kompression erreicht.

*Leistungsanpassung*. Standardisierte Codecs sollen meist möglichst breit einsetzbar sein. Dazu müssen sie nicht nur ein breites Spektrum an Medienqualitäten abdecken, sondern an Systemkomponenten unterschiedlicher Leistung anpaßbar sein. Sie sollen auch mit Software-Codecs auf langsamen PCs oder über langsame Internet-Verbindungen nutzbar sein. Ein Beispiel: Wie in Kapitel B4.3.2 beschrieben, beruht JPEG auf Kompression im Frequenzbereich mittels DCT. Um dem Betrachter einen Grobeindruck von Bildern zu erlauben, während die Dekompression oder die Übertragung noch im Gang ist, wird der sog. *progressive* Modus in zwei Varianten angeboten: während bei der einfachen Variante ein 8·8-Block des Bildes nach dem anderen DCT-transformiert (sowie quantisiert) und danach entropiecodiert wird, werden bei der „spektralen“ Variante für das gesamte Bild zunächst die niederen Frequenzen der DCT entropiecodiert und abgespeichert, dann gruppenweise immer höhere. Bei der „sukzessiven“ Variante werden alle Frequenzen gleichbehandelt, aber zuerst die höchstwertigen Bits jedes Koeffizienten entropiecodiert, dann gruppenweise niederwertigere. Daneben gibt es noch den sog. *hierarchischen* Modus, bei dem in verschiedenen Dateien verschiedene Auflösungen des Bildes gespeichert werden. Die Dateien für höhere Auflösungen bauen auf den gröbereren auf. Ein vergleichbares Prinzip kommt auch bei dem von Kodak patentierten Kompressionsverfahren für Foto-CDs zum Einsatz. Auch skalierbares Video, wie im folgenden Abschnitt be-

geschrieben, ist ein wichtiger Ansatz zur Leistungsanpassung. Audio- und Videocodierung für stufenlose Anpassung zwischen Minimal- und Maximalauflösung im laufenden Betrieb ist noch im Forschungsstadium.

**Standards für digitales Video.** Digitales Video, gegebenenfalls mit begleitendem Audio, stellt als Einzelmedium die höchsten Anforderungen an die Multimediaetechnik und soll deshalb genauer behandelt werden. Die ISO-Standards der MPEG-Serie spezifizieren Parameter, Arbeitsweise und Datenstrukturen von Audio-/Video-Codern. In der sog. Systemdefinition spezifizieren sie die Integration in ein Dateiformat bzw. in einen Transportstrom zur Übertragung. Die H.26x-Standards der International Telecommunications Union ITU – H.261 und H.263 – beschränken sich auf Video. Für Audio und die Integration in Telefonnetz-basierte Videokonferenzen gelten weitere Normen, wovon H.320 die übergreifende ist. Qualitätsunterschiede zwischen Digitalvideos innerhalb eines Standards betreffen die Parameterwahl und vor allem den codiererseitigen Aufwand für die Suche nach Bewegungsvektoren, da das Suchverfahren nicht genormt ist. Dieser Aufwand bestimmt für die DCT-basierten MPEG- und H.26x-Codierer auch den Grad der Asymmetrie (der Kompressionsaufwand übersteigt den Dekompressionsaufwand). Für Videoverteilungsnetze (und damit MPEG) ist Asymmetrie erwünscht, für Videokonferenzen (damit H.26x) unerwünscht.

**Die H.26x-Serie.** H.261-Codierer beschränken die Suche auf einen kleinen Radius oder suchen gar keine Bewegungsvektoren; zudem sind nur I- und P-, keine B-Frames definiert. Diese Beschränkungen hat H.263 nicht. Bitströme mit verfahrensbedingt schwankender Datenrate werden üblicherweise zu einem isochronen Strom geglättet, indem ein sendeseitiger Puffer bei Erreichen eines Schwellwertes gröbere Quantisierung anfordert. H.263 ermöglicht hier zusätzlich durch hierarchische Codierung mit optionalen Teilen den Umstieg auf gröbere Auflösungen. H.263 unterstützt nicht nur QCIF und CIF, sondern den Umstieg auf bis zum Sechzehnfachen von CIF. Als jüngstes unter den DCT-basierten Kompressionsverfahren ist es auch das ausgefeilteste. Daher ist es außer für Videokonferenzen auch für gespeicherte Videos, vor allem im Internet, verbreitet.

**Die MPEG-Serie.** Der MPEG-1 Standard (oft auch nur MPEG) wurde in Teilen 1992 verabschiedet; er erlaubt Kompression in Videorecorder-Qualität mit der Übertragungsrate der ersten CD-ROMs (1.2 MBit/s). Dort sind jedoch nur hohe Bildwiederholraten (24 bis 60 Hz) definiert. Die Verzögerung für Übertragung und Decodierung darf bei MPEG bis zu einer Sekunde betragen, und Codierer müssen überhaupt nicht in Echtzeit arbeiten, so daß sich dieser Standard nicht ohne weiteres für Videokonferenzen eignet.

Der MPEG-Videostrom ist hierarchisch in sechs Ebenen gegliedert.

- Ebene 1: Die kleinste Einheit bilden die entropiecodierten 8-8-Blöcke.
- Ebene 2: Darüber liegen *Makroblöcke* aus 4 Blöcken der Luminanz-(Y-)Ebene, den zu diesem 16-16-Feld gehörenden Bewegungsvektoren (0 – 2 je nach Frame-Typ) sowie je einem Block der unterabgetasteten U- und der V-Ebene (zu diesem „subsampling“ genannten Zusammenhang siehe Kapitel B4.3.3).
- Ebene 3: Hier werden Makroblöcke zu sog. *Slices* zusammengefaßt und mit übergreifenden Parametern wie Quantisierungstabellenkennungen versehen.
- Ebene 4: Hier werden mehrere Slices zu einem Einzelbild (picture) zusammengefaßt und mit Verwaltungsdaten wie der eindeutigen Bildnummer innerhalb des Videos versehen.
- Ebene 5: Hier werden Bilder zu Gruppen zusammengefaßt (*group of pictures, GOP*); GOPs enthalten üblicherweise die Abfolge von einem I-Frame zum nächsten, also z.B. die weitverbreitete IBBPBBPBB-Folge. Da B-Frames im allgemeinen nicht ohne den

nachfolgenden I- oder P-Frame decodiert werden können, wird ...  $B_{i-1}B_{i-1} | I_i B_i P_i | B_i B_i P_i^2 | B_i B_i | I_{i+1} \dots$  umsortiert zu ...  $I_i B_{i-1} B_{i-1} | P_i^1 B_i B_i P_i^2 | B_i B_i | I_{i+1} B_i B_i | \dots$

- Ebene 6: Die oberste Ebene wird *Sequence-Layer* genannt; eine Sequenz umfaßt dabei eine Menge von GOPs, für die dem Decodierer vorab in den Sequenz-Verwaltungsdaten die gemeinsame (geglättete) Bitrate und der vorzuhaltende Puffer mitgeteilt wird. Der so hierarchisierte Videostrom wird in der MPEG-Systemdefinition mit den Audioströmen zu „Packs“ verschränkt und mit Zeitstempeln versehen.

Für MPEG-2 sind Werte zwischen 4 und 100 MBit/s typisch. Da auch hier das in Kapitel B4.3.3 beschriebene Grundprinzip für DCT-basierte Videokompression angewendet wird, bewirkt die höhere Datenrate offensichtlich höhere Videoqualität. Verschiedene MPEG-2-„Levels“ bezeichnen dabei im wesentlichen unterschiedliche Auflösung (z.B. 720-572 Pixel, was sehr guter Fernsehqualität entspricht, oder HDTV-Formate mit bis zu 1920 Pixeln pro Zeile, im klassischen Verhältnis 4:3 oder im Breitwand-Format 16:9). Die wichtigste Erweiterung gegenüber MPEG-1 betrifft skalierbare Videoströme, die in den sog. MPEG-2-Profilen geregelt werden. Interessant ist dabei die räumliche Skalierung, die der für JPEG beschriebenen hierarchischen Codierung ähnelt. Damit kann z.B. ein HDTV-Video unterabgetastet und einer 4-4-DCT zugeführt werden, die 8-8-DCT kann dann differentiell darauf aufgebaut werden. So kann sich ein Decodierer für normale Fernseher auf die Decodierung der 4-4-Anteile beschränken, normale und HDTV-Fernseher können über denselben Videostrom bedient werden.

**Internet-Video und Digitales Fernsehen.** Teilstandards von MPEG-4 werden seit 1999 verabschiedet. Ursprünglicher Schwerpunkt war die Unterstützung niedriger Bitraten, viele der Optimierungen gegenüber MPEG-1 und -2 taugen aber auch für höhere Raten, insbesondere wenn auf den im Fernsehen üblichen Halbbild-Modus verzichtet werden kann. MPEG-4 wird deshalb und wegen der in Abschnitt 2.3.3 beschriebenen Erweiterungen zunehmend allgemeiner Internet-Videostandard. Der Begriff „streaming video“ bezeichnet die Möglichkeit, während der Übertragung mit dem Abspielen zu beginnen. Mangels Dienstgüteunterstützung im Internet-Protokoll IP muß auf schwankende Netzlast mit Ratenanpassung und Empfangspuffer reagiert werden. Microsoft, Real und Apple liefern sich einen Kampf um die beste Lösung. Microsoft Media mischt MPEG-4 mit einem eigentumsgeschützten Audiocodec, Quicktime favorisiert neben MPEG-4 den nichtstandardisierten Sorensen-Codec, der auf Vektorquantisierung beruht (siehe Kapitel B4) und Real baut auf völlig eigentumsgeschützte Lösungen. Mit der Internet Streaming Media Alliance ISMA versuchen Mitgliederfirmen, MPEG-4 als offenen Streaming-Standard voranzutreiben. Als Kommunikationsprotokoll oberhalb von IP und UDP setzt sich das Real-Time Streaming Protocol RTSP durch, die in IP eigentlich vorgesehene Multicast-Unterstützung bedauerlicherweise noch nicht.

MPEG-2 ist als Fernseh- und HDTV-tauglicher Standard für DVDs, Digitalfernsehen und zunehmend Camcorder etabliert. Digitalfernsehen wird in Europa von der DVB (digital video broadcasting) vorangetrieben und in Nordamerika von ATSC und der Herstellervereinigung SMPTE. Kern ist jeweils MPEG-2, regionale Interessen der Vereinigungen und Regulierungsbehörden verhindern eine weltweite Norm. DVB kennt Unterstandards für Kabel, Satellit und terrestrische Ausstrahlung (DVB-T), die Ablösung analoger Standards ist z.B. in Deutschland per Gesetz terminlich festgelegt. Für die sog. „Set-Top-Box“, also das Empfangsgerät, das haushaltsübliche Fernsehgeräte (engl. TV-Set) Digitalfernsehtauglich macht, beziehen die DVB-Normen Java mit ein und erlauben somit dynamisch ladbaren Code – der Unterschied zwischen Fernsehgerät und Computer wird verwischt.

### 2.3.3 Entwicklung und Ausführung

Wie in Abschnitt 2.1 erwähnt, entwickeln sich Mikrowelten und Frameworks im Funktionsumfang aufeinander zu. Als Entwicklungsumgebungen und Programmierschnittstellen für Anwendungsentwickler (APIs) sind vor allem Apples Quicktime, Microsoft's DirectX und Sun's Java Media Framework (JMF) verbreitet, siehe unten. Zunächst wird nochmals auf MPEG-4 sowie MPEG-7 und MPEG-21 eingegangen, die hier ebenfalls eine Rolle spielen.

Die MPEG-4-Unterstandards umfassen mehr Bereiche als die bereits erwähnten, insbesondere Interaktivität, Mischung realer und computergenerierter Medien sowie Synthetisierung von Medienströmen. Für Videokonferenzen kann ein 2D-Netz über das „neutrale“ Sprecher-Bild gelegt werden; Lippenbewegungen und Mimik werden dann als Punkteverschiebungen in diesem Netz modelliert, die ganze Videokonferenz wird so zu einer Folge solcher Verschiebungen komprimiert. Auch 3D-Modelle des Kopfes mit dem Gesicht als Textur werden unterstützt, als Teil einer allgemeinen Beschreibungssprache XMT für animierte 3D-Szenen (VRML in der Fassung von 1997, ergänzt um einen Mechanismus zur Spracherweiterung und um eine kompakte binäre Darstellung BIFS).

Synthetisierung sei am Beispiel der Fernsehübertragung eines Tennisspiels erläutert. Dort könnte man sich vorstellen, den Hintergrund (Platz, Zuschauer) als Panoramabild mit geringer Wiederholrate (z.B. mit Wavelet-Kompression) zu übertragen und die aktuelle Kameraeinstellung als Parameter des darzustellenden Bildausschnittes laufend zu senden. Der Tennisspieler mit Schläger könnte als freigestelltes Objekt ohne Hintergrund mit normaler Wiederholrate, aber geringerem Datenvolumen gesendet werden. Der Ball könnte ebenfalls freigestellt und mit hoher Wiederholrate, aber auch als mathematischer Ausdruck codiert werden. Im Empfänger würden die drei Ebenen zusammengefügt. Weiter stelle man sich mehrere parallel übertragene solche Multi-Ebenen-Ströme vor, die über ein Kabelnetz an viele Haushalte gehen. Per Schaltfläche (als weitere Ebene in das synthetisierte Fernsehbild integriert), könnte der Fernsehzuschauer den Wechsel zu einem anderen Strom d.h. einer anderen Kameraposition anfordern. Kommentare mit unterschiedlichen Schwerpunkten oder Sprachen, Platzgeräusche und Hintergrundmusik könnten vom Zuschauer ebenfalls interaktiv und individuell gemischt werden.

In MPEG-4 ist das beschriebene Szenario über die „MPEG-4 syntactic description language“ MDSL und das Konzept audiovisueller Objekte (AVO, im Beispiel Panoramabild, Spieler, Schaltfläche, Kommentar usw.) realisierbar. Selbst die Frage, wie Urheberrechte an AVOs durch nicht-löschbare Signaturen geschützt werden können, wird behandelt. Zu den weiteren Komponenten zählen die intervallbasierte Sprache *FlexTime* zur Synchronisationsspezifikation, *DMIF* (*delivery multimedia integration framework*) zur Unterstützung von Medienströmen im Netz und MPEG-J als Java-Schnittstelle für die Einbettung interaktiver Anwendungsprogramme.

Für Studios und Sender wird als Teil von MPEG-4 ein Werkzeugkasten angeboten. Er enthält Funktionen wie die Extraktion und Bewegungsschätzung freizustellender Objekte aus dem Fernsehkamera-Input. Dieses Problem ist nach heutigem Stand der Technik allerdings noch nicht befriedigend gelöst; man kann sich in zukünftigen Studios aber sehr wohl einen Satz virtueller digitaler Requisiten als AVOs vorstellen und daran ermesen, daß MPEG-4 der Filmproduktion neue Wege weist.

Da die Arbeiten an MPEG-3 durch MPEG-2-Erweiterungen überholt und daher eingestellt wurden, wollten die Verantwortlichen mehr zum Spaß Numerierungslücken nach mathematischen Regeln einführen (+1, +2, +3, ...) und bezeichneten daher den nächsten Standard als MPEG-7. Mit dem nachfolgenden MPEG-21 (für „21. Jahrhundert“) wurde diese Regel aber bereits wieder durchbrochen.

MPEG-7 konzentriert sich auf das Problem der Inhaltsanalyse und -beschreibung von Medienströmen, die sog. Metadaten. Konkurrierende Bestrebungen für Hypermedia werden in Abschnitt 2.5.4 und B4.3.4 beschrieben. MPEG-7 wird eine Beschreibungssprache standardisieren, mit der der syntaktische Aufbau von Metadaten beschrieben werden kann. Möglicherweise wird für wichtige Anwendungsbereiche außerdem eine Ontologie, d.h. eine Art einheitlicher Thesaurus von Schlüsselwörtern genormt, mit dem Inhalte beschrieben werden können. Allerdings sind viele Ebenen der Beschreibung denkbar, für Video z.B. die Beschreibung von Kamerabewegungen, Einstellungen (wie „long shot“ für eine ganz bestimmte Größenrelation von Sprecher zu Hintergrund), Form und Beschaffenheit geometrischer Objekte (gelbes Trapez) oder realer Objekte (Auto von links), handlungsspezifische Szenen und Objekte (Marylin küßt), usw., so daß Aufwand und Erfolgsaussichten für solche Bemühungen strittig sind. MPEG-7 kümmert sich auch um die Integration der Metadaten in Medienströme (vgl. digitale Sender- und Sendungstypkennungen bei RDS-Radios) und um einheitliche Schnittstellen zu Autorenwerkzeugen, Suchmaschinen usw. MPEG-21 soll Rahmenstandard für die bisherigen werden und alles ergänzen, was für rechnergestützte Produktion und elektronischen Handel mit interaktiven digitalen Medien noch fehlt.

Die vorstehende Beschreibung macht deutlich, daß MPEG als Entwicklungs- und Laufzeitunterstützung verstanden werden kann. Microsoft unternimmt mit DirectX große Anstrengungen, die früher getrennte Unterstützung für Spiele, Grafik, Video usw. zu vereineitlichen als DirectPlay, DirectDraw/Direct3D, DirectShow usw. Neben der Kapselung verschiedener Codecs, z.B. im AVI-Format des Medientyps VideoForWindows, bietet es ein sog. Hardware Abstraction Layer HAL, das Geräteunterschiede wie die von Grafikkarten gegen Anwendungsprogramme abschirmt: nicht im Gerät oder der Einschubkarte verfügbare Funktionen werden in Software emuliert. In DirectShow werden Softwaremodule für die Verwaltung von Medienquellen, zum Mischen, Bearbeiten und Filtern von Strömen sowie zur Präsentation in wenige sog. Filterklassen zusammengefaßt und zu Filtergraphen konfiguriert. Die sog. DirectX Media Objects DMO verallgemeinern dieses Konzept für alle DirectX-Komponenten und bieten eine vereinfachte Programmierschnittstelle.

Quicktime arbeitet mit dem universellen Datentyp *movie*, der beliebige Medien als sog. *tracks* (Spuren) synchronisiert. Die sog. *movie\_toolbox* steht als universelle Abspiel- und Editierkomponente anderen Anwendungen zur Verfügung. Der *compression manager* verwaltet die verfügbaren Kompressionsverfahren und Dateiformate, und der erweiterbare *component manager* verdeckt Details von Software- und Hardwaremodulen für Kompression, Aufnahme und Präsentation. Quicktime unterstützt Echtzeitanforderungen nur rudimentär, die konsequent objekt- und komponentenorientierte Architektur ist im übrigen vorbildlich.

Das Java Media Framework JMF wird seit der Version 2.0 vor allem von Sun und IBM vorangetrieben. Es unterstützt mehr Betriebssysteme als Apple und Microsoft, beschränkt sich andererseits auf ein einheitliches Modell und die Programmierschnittstelle zur Systemsoftware, liefert also selbst keine Werkzeuge mit. Eine mit Quicktime vergleichbare Plugin-Schnittstelle „nach unten“ erlaubt die Erweiterung um Verarbeitungshard- und software wie Codecs, Multiplexer usw. Wie bei DirectX werden wenige Modulklassen unterschieden wie Player, Processor, Medienquelle und -senke. Für deren Konfiguration wird das von JavaBeans bekannte ereignisbasierte Komponentenmodell verwendet. Zeitkapseln werden mit Hilfe der Klasse *clock* realisiert.

### 2.3.4 Speicherung

Auf die ausgiebige Behandlung der Multimedia-Speicherung muß hier verzichtet werden. Unter den Speichermedien hat die CD-ROM-Familie besonders große Bedeutung, da verschiedene Unterarten (CD-ROM/XA, CD-I, CD-Video, DVD als Nachfolger) speziell für Multimedia ausgelegt wurden: dies betrifft sowohl spezielle Dateiformate (z.B. für ADPCM-codiertes Audio bei CD-I) als auch die Speicherorganisation (z.B. für die verschärfte Speicherung von synchron abzuspielenden Audio- und Videodateien in CD-ROM/XA). Als weitverbreitete Wechselmedien mit wahlfreiem Zugriff eignen sie sich besonders zum Multimedia-Datentransport (siehe Kapitel C4).

Speicherorganisationen wurden insbesondere für Video-on-Demand-Server intensiv erforscht. Dort ist vor allem die Verteilung von Videos auf Speicherhierarchien interessant, wenn z.B. Hunderte von Haushalten gleichzeitig auf dasselbe aktuelle Video zugreifen, aber auf Grund versetzter Anfangszeiten an verschiedenen Stellen. Hier reicht es meist nicht, mehrere Kopien bereit zu halten; weit verbreitet ist das sog. „striping“: das Video wird in Schnipsel zerstückelt. Die Schnipsel werden z.B. auf eine „Durchsatz-Hierarchie“ aus einem Kabelverteilsystem mit  $n$  Servern mit je  $m$  Controllern mit je  $q$  Platten verteilt, wobei aufeinanderfolgende Schnipsel auf verschiedenen Platten gespeichert werden. Dadurch wandert der einzelne Zuschauer bei seinem Zugriff zyklisch die Peripherie ab und die (einem Zufallsprozeß gleichenden) vielen Konsumenten desselben Videos verteilen sich auf die gesamte Server-Hardware. Weitere Probleme betreffen die große Zahl anbietender Videos bei extrem variierender Nachfrage, schnellen Vor- und Rücklauf und wahlfreie Zugriffe auf interaktive Videos.

In Datenbanken ist die Behandlung von Multimediadaten durch drei Problemfelder gekennzeichnet: (1) große Datenmengen, deren Behandlung als *Binary Large Objects* (BLOB) zwar schon für die Büroautomation untersucht wurde, bei Multimedia aber durch Realzeitanforderungen und gegebenenfalls spezielle Speichermedien erschwert wird; (2) variierende Operatoren für Multimediadaten – diese lassen sich nicht wie die für Basisdatentypen bekannten (arithmetischen, logischen usw.) Basisoperationen fest in ein Datenbank-Modell integrieren; (3) (semi-)automatische Deskribierung, welche in engem Bezug zur nachfolgend besprochenen Interpretation steht.

### 2.3.5 Interpretation

*Indizierung durch Medienanalyse.* In umfangreichen Forschungsarbeiten wird versucht, höhere Semantik aus der signalorientierten Medienrepräsentation zu extrahieren, z.B. durch Sprachanalyse, Handschrift-Erkennung, Bildverstehen oder automatische Erzeugung charakteristischer Sinnbilder für Video [Multimedia Systems 95]. Interpretationsaufgaben erfordern immer den zeitunkritischen Schritt der Merkmalsextraktion aus gespeicherten Mediendaten; bei a priori bekannter Fragestellung (z.B.: Sprache vollständig in Text übersetzen) kommt ein weiterer zeitunkritischer Schritt der Merkmalsanalyse hinzu, für Datenbank-ähnliche Abfragesysteme dagegen ist dieser Schritt zeitkritisch und von der konkreten Abfrage abhängig (z.B.: finde Bilder von Häusern mit grünem Verputz). Grundzüge der Bild- und Schriftanalyse werden in Kapitel E1.8 angerissen.

*Beispiel Sprache.* Für Sprachverstehen werden heute hierarchische Verfahren eingesetzt, die aus phonetischer, syntaktischer und – in Ansätzen – semantischer Analyse bestehen. Die phonetische Analyse als unterste Ebene betrachtet Phoneme als kleinste „akustische“ Einheiten bzw. Diphonge (Doppellaute) und deren Verkettungen. Sie werden mit Buchstaben bzw. Wörtern und deren zuvor gespeicherten charakteristischen Wellenformen verglichen. Wie beim menschlichen Sprachverstehen werden nicht oder schlecht entscheid-

bare Alternativen auf der nächsthöheren Ebene betrachtet („Vieh“ und „Fieber“ syntaktisch, „das Q“ und „die Kuh“ semantisch). Diese mehrschichtige Auflösung von Mehrdeutigkeiten (Disambiguierung) gelingt nur dann zufriedenstellend, wenn auch Wissen über das besprochene Themengebiet einbezogen wird; praktikable Anwendungen sind deshalb bislang stark eingeschränkt. Unter „wordspotting“ versteht man die Suche nach Schlüsselwörtern in längeren Audioaufzeichnungen zur Indizierung.

*Beispiel Video.* Die Forschung im Bereich Videoanalyse wendet sich zunehmend sehr anspruchsvollen Aufgaben zu. Nachfolgend sei beispielhaft ein kleiner Einblick in das Problem der Erkennung von Szenenwechseln gegeben, das zu den einfachsten gehört, aber noch nicht befriedigend gelöst ist. Innerhalb einer Filmszene kann erwartet werden, daß sich der Bildinhalt von einem Einzelbild zum nächsten nur graduell ändert; bei einem harten Schnitt dagegen ist mit einem drastischen Wechsel zu rechnen. Beim Bildvergleich kann der YUV-Farbraum unter Weglassung der Luminanzebene  $Y$  vorteilhaft sein, weil er plötzliche Helligkeitsänderungen durch Blitzlichtkameras (vgl. Pressekonferenz) oder Blendenänderung der Videokamera als Fehlerquelle weitgehend ausschließt. Einfache, aber wirkungsvolle Schnitt-Erkennungsverfahren arbeiten mit Farbhistogrammen als Merkmalen. Dazu werden die Millionen möglicher Farben (bzw. möglicher  $U/V$ -Werte) in  $n$  Klassen, sog. *Bins*, eingeteilt und die Pixel eines Bildes quasi in die Bins einsortiert; es entsteht ein Histogramm  $H[1 \dots n]$  vom Umfang der Bin-Anzahl. Der Vergleich zweier Bilder  $B$  und  $B'$  kann nun in der Summe der Bin-weisen Histogramm-Unterschiede bestehen:

$$\text{Diff} = \sum_{i=1 \dots n} |H[i] - H'[i]|$$

Überschreitet der Wert Diff einen bestimmten Schwellwert, so wird ein Szenenwechsel im Film (harter Schnitt) angenommen. Fehler durch sich schnell bewegende Objekte können ausgeschlossen werden, indem das Bild in mehrere (z.B. neun) Regionen zerteilt wird und ein Quorum von Quadranten einen Schwellwert überschreiten muß. Die Erkennung weicher Szenenwechsel geht davon aus, daß Überblendungen und ähnliche Effekte über eine Anzahl von Einzelbildern hinweg zwar keine abrupte Änderung, dafür aber in jedem Fall ein Mindestveränderung erzeugen. Die Diff-Werte werden deshalb aufsummiert. Wird von einem Bild zum nächsten ein unterer Schwellwert unterschritten, so wird die Summe auf Null zurückgesetzt (sehr wenig Änderung, derzeit also kein weicher Szenenwechsel). Steigt die Summe dagegen über mehrere Bilder hinweg über einen Grenzwert, so wird auf Szenenwechsel geschlossen. Die Trefferquote hängt noch stark vom Videomaterial ab, ausgeklügeltere Verfahren brachten bislang auch noch nicht den Durchbruch. Geforscht wird u.a. an Verfahren auf Basis der Autokorrelation von Farb-Bins, an Verfahren im Frequenzbereich, die z.B. für MPEG-Filme Dekompression ersparen, und am Einsatz von Kantenerkennung. Letztere ist aus der Bildverarbeitung bekannt; an die Stelle von Diff tritt als Unterschied zweier Bilder die Summe aus weggefallenen und hinzugekommenen Kanten. Kamerabewegungen müssen vor dieser Differenzbildung kompensiert werden, was nur teilweise gelingt. Bei MPEG können Art und Umfang von Kamerabewegungen aus den Bewegungsvektoren abgeleitet werden; die Ergebnisse hängen aber stark von der Qualität des Codierers ab, mit dem die Videodaten erzeugt wurden.

*Indizierung bei der Entstehung.* Die zunehmende Digitalisierung von Endgeräten und Allgegenwart der Computernetzung eröffnen zunehmend Möglichkeiten, Mediensemantik bei der Entstehung der Daten zu erfassen. So stellt sich beispielsweise die Frage, ob automatische Inhaltsbeschreibungen für Spielfilme mit (rechenintensiven, bislang wenig leistungsfähigen) Szenenanalyseverfahren erzeugt werden sollten, wenn in zukünftigen digitalen Filmstudios maschinenlesbare Drehbücher viel eleganter für diesen Zweck



eingesetzt werden können. Auch hier stellt sich allerdings das Problem der Metadaten-Beschreibung und -Standardisierung (Ontologien) wie im Zusammenhang mit MPEG-7 beschrieben.

## 2.4 Systemunterstützung

### 2.4.1 Netzwerke und Kommunikationsdienste

Um isochrone Medienströme zuverlässig zu unterstützen, müssen für den gesamten Datenpfad vom Ursprung (live-Aufnahme, Lesen vom Speicher) bis zur Senke (Lautsprecher, Bildschirm usw.) lückenlos Dienstgütereigenschaften gegeben werden. Heutige Multimedia-PCs und Telemedia-Anwendungen berücksichtigen dies meist nicht, sondern beruhen auf der Hoffnung, daß keine Ressourcenknappheit auftritt.

An der Schnittstelle zwischen Anwendungsprogramm und Netz werden nicht nur zusätzliche Dienstgüte-Parameter relevant, zum Teil verschiebt sich auch die Bedeutung:

- *Durchsatz* (z.B. implizit definiert durch die Abtast-/Bildwiederholrate, Auflösung usw.), früher Hauptanliegen der Netzbetreiber, wird wichtige Anwender-Forderung.
- Die *Bitfehlerrate* sinkt auf optischen Medien beträchtlich, Anwendungen sind zum Teil weniger anfällig (vgl. Pixelfehler in Bildern mit Bitfehlern in ausführbaren Dateien), dagegen steigt in schnellen zellenvermittelten Netzen wie ATM die *Zellenverlustrate*. Bei komprimierten Multimedia-Daten, z.B. gemäß MPEG, variiert die Fehlerempfindlichkeit: sie sinkt von grundlegenden Verwaltungsdaten über I- und P- bis zu B-Frames. Traditionelle prüfsummenbasierte Fehlerbehandlung mit Übertragungswiederholung verzögert die Weiterleitung des Mediendatenstroms, bis das „Reparaturpaket“ eingetroffen ist: isochroner Datenverkehr ist so kaum zu erreichen. Daher wird Redundanz eingefügt, die die Rekonstruktion ganzer Pakete ohne Übertragungswiederholung erlaubt (Vorwärts-Fehlerkorrektur), idealerweise mit variablem Sicherheitsgrad.

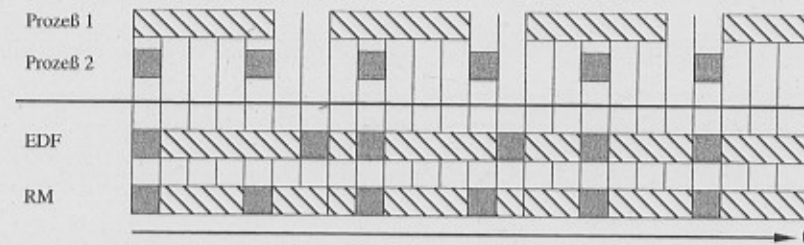
Multimedia-Kommunikation erfordert zunächst auf den unteren Schichten isochronen Datenverkehr und garantierten Durchsatz. Das ist in lokalen Netzen mit Mehrfachzugriff im zufallsgesteuerten Wettbewerbsbetrieb (Ethernet) nicht möglich. Auch auf höheren Schichten und bezüglich der gesamten Schichtenstruktur des OSI-Referenzmodells treten Probleme im Zusammenhang mit Hochgeschwindigkeitsnetzen und Multimedia auf (siehe Kapitel D10). Durch strenge Schichtung und die große Anzahl von Schichten entsteht insbesondere ein Software-Engpaß, bei dem Funktionen sequentiell, zum Teil mehrfach oder unnötig ausgeführt werden. Parallelisierung, Zusammenfassung von Schichten sowie benutzergesteuerte Auswahl von Funktionseinheiten sind hier die wichtigsten diskutierten Gegenmaßnahmen. Für transportorientierte Schichten werden neue und erweiterte Protokollmechanismen untersucht, vor allem die folgenden [Fluckiger 95, Heinrichs 96]:

- Effizientes, zuverlässiges Gruppensenden (Multicast) mit minimaler Duplizierung von Paketen und geeigneter Fehlerbehandlung, als Ergänzung der klassischen Verfahren Unicast (wo z.B. pro Empfänger eine separate Videoverbindung hergestellt wird, obwohl sich viele Empfänger am selben entfernten Teilnetz befinden) und Broadcast.
- Alternative Fehlerbehandlungsmaßnahmen wie oben erwähnt.
- Impliziter Verbindungsaufbau, Ratenkontrolle: im Hochgeschwindigkeits-Weitverkehr (z.B. 500 MBit/s, 1000 km) ist das Signal für ein Bit relativ kurz (im Beispiel rund 40 cm). Eine Datei (1 MBit: 400 km „Bits“) könnte daher komplett abgesendet

sein, bevor auch nur ein kurzes Verbindungsaufbauwunsch-Paket beim Empfänger ankäme. Statt des klassischen Verbindungsaufbaus durch mehrfache Paketübertragung (Handshake) trägt daher das erste Datenpaket Verbindungsdaten. Weitere Pakete werden sofort übertragen; statt der Anzahl unquittierter Pakete (Flußkontrolle) ist nur die *Paketrate* begrenzt.

### 2.4.2 Betriebssysteme

Die wichtigste Multimediaerweiterung für Betriebssysteme betrifft die geeignete Prozessorzuteilung [Steinmetz 95]. Bild 2 illustriert zwei wichtige Verfahren hierfür.



**Bild 2** Prozessorzuteilung nach EDF und RM  
Bei Prozeß 1/2: Rechtecke zeigen Anforderungen von Prozessorzuteilung.  
Bei den Algorithmen: Rechtecke zeigen die zugehörige Prozessorzuteilung an.

Der Earliest-Deadline-First-(EDF-)Algorithmus teilt aus der Bereitmenge immer den Prozeß mit der frühesten Frist (deadline) ein. EDF ist optimal in dem Sinne, daß ein geeigneter Plan gefunden wird, falls einer existiert. Da bei jedem Eintritt eines Prozesses in die Bereitmenge neu geplant werden muß, hat er eine hohe Berechnungskomplexität und erscheint für kontinuierliche Medien mit deterministischen Betriebsmittelanforderungen überdimensioniert. EDF ist auch für klassische Realzeitprobleme geeignet.

Ratenmonotone (RM) Planung basiert auf einschränkenden Annahmen: (1) pro Prozeß fällt die Frist einer Bearbeitung mit der Ankunft der nächsten Anforderung zusammen (für Videoströme: ein Bild muß dargestellt sein, bevor das nächste eintrifft); (2) nur die Bearbeitung kontinuierlicher Medienströme ist zeitkritisch; (3) pro Bearbeitung wird dieselbe Bearbeitungszeit angefordert (bei komprimierten Medien variabler Bitrate nicht gegeben). Unter diesen Bedingungen genügt es gemäß RM, jedem zeitkritischen Prozeß eine ratenabhängige Priorität zuzuordnen (je höher die Rate, desto höher), zeitunkritischen Prozessen die niedrigste. Eine Prozeßmenge ist fristgerecht planbar, falls der ungünstigste Fall befriedigt werden kann: für den zeitkritischen Prozeß niedrigster Priorität trifft eine Anforderung zeitgleich ein mit Anforderungen für alle anderen Prozesse. Im Gegensatz zu EDF muß bei RM eine Prozeßmenge nur einmal geplant werden. Dafür ist hundertprozentige Prozessorauslastung bei Vollast nicht garantiert.

## 2.5 Hypertext / Hypermedia

### 2.5.1 Grundbegriffe

Papierbasierte textuelle Information ist inhärent sequentiell. Nichtsequentielle Strukturen und Beziehungen wie Hierarchien (Kapitel), Fußnoten und Querverweise müssen textuell ausgedrückt werden und behelfen sich mit sekundären Ordnungsmerkmalen wie Seitenzahlen, was nichtsequentielles Lesen sehr erschwert (Blättern). Grundidee des *Hypertext*-Konzeptes ist es, nichtsequentielle Strukturen in computerbasierten Dokumenten sichtbar zu machen. Die zentralen Begriffe sind dabei *Knoten*, *Link*, *Anker* und *Netz* [Nielsen 95]. Die Information wird in kleine semantisch zusammenhängende Informationseinheiten, sog. *Knoten (nodes)*, zerlegt. Beziehungen jeder Art zwischen diesen Knoten werden durch explizite Verweise, sog. *Links*, repräsentiert. Wesentlich ist die Möglichkeit, Links über eine leistungsfähige, meist graphische Benutzeroberfläche leicht aktivieren zu können. Meist beginnen und enden Links an bestimmten Ausschnitten eines Knotens, z.B. Stichworten, Teilsätzen, im Falle graphischer Knoten auch an dargestellten Objekten. Diese Ausschnitte werden als (Quell-, Ziel-)Anker bezeichnet. Ein vollständiges Dokument wird bei der beschriebenen Vorgehensweise zu einem zusammenhängenden Graphen aus Knoten und Links. Hierfür hat sich bisher keine einheitliche Bezeichnung durchgesetzt, es wird u.a. *Hypertext*, *Hypertext-Dokument*, *Web* und wie nachfolgend *Netz* genannt.

Bisweilen wird die in Abschnitt 2.1 erwähnte Karteikarten-Metapher übernommen. Knoten haben dann einheitlich Kartenformat; Zielanker entfallen meist, da Links zu ganzen Karten führen. Der Begriff *Hypermedia* wird wie nachfolgend verwendet, wenn die Verwendung mehrerer, insbesondere zeitabhängiger Medien betont werden soll.

Für das *Lesen* von Hypermedia-Dokumenten sind die Begriffe *Stöbern*, *Navigieren* und *Abfragen* relevant. *Stöbern (browsing)* entspricht dem Blättern in Papierdokumenten mit dem Ziel, einen Überblick zu gewinnen und relevante Knoten zu finden. Es wird nachhaltig unterstützt durch eine intuitive Repräsentation des Netzes als Graph und ist daher ein wichtiges Anwendungsfeld des Graphenlayout. Charakteristische Beschreibungen der Knoten bzw. Illustrationen (*Icons*) sind wichtig zur Inhaltsabstraktion. *Navigieren* bezeichnet planmäßiges Durchlaufen und Lesen von Knoten und ähnelt dem Lesen in Papierdokumenten. *Abfragen*, ein weniger breit akzeptierter Begriff, soll für die Formulierung sog. *Hypertext-Queries* stehen, die inhalts- und strukturorientierte Aspekte verknüpfen. So könnte eine einfache Abfrage wie

```
<node where author != 'X'> <link> [<node><link>] * <node where author = 'X'>
```

alle direkten und indirekten externen Verweise auf Dokumente des Autors X suchen; *author* ist als inhaltsorientiertes Attribut außerhalb des Systemmodelles zu verstehen.

### 2.5.2 Das Dexter-Modell

Das sog. *Dexter-Modell* (Bild 3) dient als Referenzarchitektur für Hypertextsysteme [CACM 94]. In seiner *Laufzeitschicht* wird die Interaktion mit dem Hypertext-Leser abgehandelt, und die von ihm initiierten Operationen werden koordiniert und ausgeführt bzw. weiterdelegiert. Die *Speicherschicht* verwaltet und speichert alle Links als Quell- und Zielknoten-Identifikation sowie Ankerinformation, nicht aber die Knoteninhalte. Für jeden Medientyp (im weitesten Sinne, siehe nächster Abschnitt) ist eine sog. *Komponente* zuständig, in der *Komponentenschicht* werden also die Knoteninhalte gehalten.

Da im Kontext verschiedener Medientypen (oder gar Anwendungen) gegebenenfalls verschiedene Ankertypen verwendet werden, werden diese in der Komponentenschicht definiert und in der Speicherschicht verwaltet. Sollen z.B. in editierbarem Text Anker an eine Textauswahl gebunden werden, auch wenn diese im Text verschoben wird, so muß der Anker selbst im Text gespeichert und nur eine (eindeutige) Ankeridentifikation in der Speicherschicht gehalten werden. Sollen dagegen Grafiken in einem Standardformat gespeichert werden, das keine Anker kennt, so muß die Ankerinformation in der Speicherschicht gehalten werden. Will der Leser interaktiv einem Link folgen, so übergibt die aktuelle Komponente die Quellanker-Identifikation der Speicherschicht. Diese ermittelt den Link und damit die Zielanker-Identifikation und übergibt diese der Komponentenschicht (gegebenenfalls einer anderen Komponente). Zielknoten und -anker werden dargestellt. Komponentenuabhängige Details der Darstellung (Benutzerpräferenzen, Gesamtlayout usw.) werden per Darstellungsspezifikationen zwischen Speicher- und Laufzeitschicht ausgetauscht.

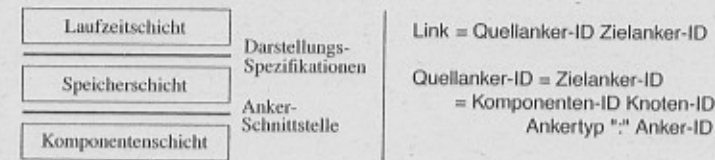


Bild 3 Dexter-Referenzmodell; Definition für Link und Anker in EBNF

### 2.5.3 Weitergehende Aspekte

*Offene Linksysteme.* Es erscheint sinnvoll, beliebige Softwarewerkzeuge verschiedener Hersteller in ein Hypermediasystem einzubeziehen. Dabei können als Knoten so unterschiedliche Artefakte vorkommen wie Kalendereinträge eines Planungswerkzeugs, E-mails oder Quelltexte eines syntaxgesteuerten Editors. Daher muß neben der Ankertypdefinition auch die Darstellung der Artefakte den Werkzeugen überlassen werden. Mechanismen sind anzubieten, um die Benutzerschnittstelle der Werkzeuge um Operationen zu erweitern wie <definiere Selektion als Quell-/Zielanker>, <Verfolge Link> usw. und die Interaktion mit dem eigentlichen Hypertextsystem abzuwickeln. Dieses ausdrucksstarke Konzept fand bislang mangels breit akzeptiertem Standard leider wenig Verbreitung.

*Wissensverarbeitung.* Hypertext-Konzepte eignen sich zur Wissensrepräsentation mit semantischen Netzen, falls benutzerspezifische Knoten- und Linktypen sowie möglichst verschiedene Abstraktionsebenen von Netzen unterstützt werden.

*Dynamik.* Dynamische Knoten (ausführbarer Code), Links (beim Verfolgen wird ein Codestück ausgeführt, welches Zielanker berechnet oder Variablen aktualisiert) und Netze (Knoten und Links können zur Laufzeit hinzugefügt oder entfernt werden) können die Verschmelzung von Autoren- und Softwareentwickler-Tätigkeit unterstützen (siehe Abschnitt 2.1).

*Verteilung.* Die Internet-Anwendung World-Wide Web (WWW) trennt Benutzerschnittstelle (browser) und Knoten-/Link-Verwaltung nach dem Client-/Server-Konzept. Verteilte Netze werden nur insofern unterstützt, als Knoten beliebige Linkinformationen vollständig enthalten, wodurch Clients bei Interpretation eines Knotens Links zu anderen Servern anbieten können. Fortgeschrittenere Konzepte umfassen dagegen Konsistenz-

haltung bei Veränderung von Zielknoten oder -ankern, Ortstransparenz durch Verwendung von Namensdiensten und Verteilungstransparenz mit automatischer Migration oder Replikation usw. Kooperatives Arbeiten wird bislang zu schlecht unterstützt: meist fehlen Konzepte für Benutzerrollen und -rechte, kooperative Arbeitsabläufe usw. Klassische Datenbankansätze eignen sich hier nur bedingt auf Grund inadäquater grundlegender Eigenschaften von Transaktionen, vor allem Kurzlebigkeit und Isolation statt Kooperation.

**Multimedia-Integration.** Typische Hypermediasysteme erfüllen nicht die Multimedia-Definition gemäß Abschnitt 2.1, da sie die Synchronisation mehrerer, auch zeitabhängiger Medien nicht unterstützen. Meist kann für synchronisierte Multimediadarstellungen ein spezieller Medientyp eingeführt werden, doch kontrolliert dann die zugehörige Komponente diese Darstellungen, die Einzelmedien bleiben dem Hypermediasystem verborgen. Im Gegensatz dazu integriert das sog. *Amsterdam-Modell*, eine Dexter-Erweiterung, Hypermedia- und Synchronisationskonzept. Synchronisierte Knoten werden zu Metaknoten zusammengefaßt, bleiben aber in ihrer Identität erhalten. Der Benutzer muß nur mit einem System umgehen, er kann denselben Knoten innerhalb und außerhalb der Präsentation verwenden, und er kann mit Links direkt in die beteiligten Medien hinein verweisen. Für das World-Wide Web wurde jüngst eine einfache Skriptsprache zur Synchronisationsbeschreibung namens SMIL (sprich: „smile“) verabschiedet.

## 2.5.4 Das World-Wide Web als Hypertextsystem

**Einordnung.** Der Name des Kommunikationsprotokolls zwischen WWW-Clients (Browsern) und -Servern; „hypertext transfer protocol (http)“, kennzeichnet WWW als Hypertext-System. Korrekter kann man es als Standard für Internet-weit zugängliche Textdokumente bezeichnen mit schrittweise verbesserter Hypertext- und Multimedia-Unterstützung. Die enorme Verbreitung ist also mehr auf die breite Akzeptanz im Internet zurückzuführen als auf ausgefeilte Hypertext-Funktionalität. Die meisten Knoten sind Textdokumente im sog. Hypertext-Markup-Language- (HTML-) Format (siehe Kapitel E8.3.5. In Markup-Sprachen wird die Textformatierung durch in den Text eingestreute Kommandos, die Markups, gesteuert (Gegensatz: WYSIWYG-Dokumentverarbeitung wie in Microsoft Word). HTML-Markups werden als *Tags* bezeichnet und kennzeichnen üblicherweise Anfang und Ende eines zu formatierenden Elementes (z.B. <title> Dies ist ein Titel </title>).

WWW befindet sich stark im Umbruch durch vielversprechende Standards wie die „*extended markup language XML*“, das „*resource description framework, RDF*“, die Link-Spezifikationsprache Xlink und SMIL für Multimedia-Synchronisation [W3C]. Vielfalt, Komplexität und teilweise Überlappung der Ansätze lassen die Entwicklung ebenso offen erscheinen wie das begrenzte Durchsetzungsvermögen des Standardisierungsgremiums W3C gegenüber der Firma Microsoft, die zunehmend den Browser-Markt beherrscht.

**WWW als Textverarbeitungssystem.** Betrachtet man WWW als Markup-Textsystem, so hat HTML Schwachstellen. Zunächst fällt auf, daß HTML direkte Layout-Anweisungen (Fettschrift, Font-Auswahl, Hintergrundfarbe) nicht von logischen Formatierungshinweisen (Überschrift, Hervorhebung) trennt. Hochwertige Markup-Textsysteme wie der ISO-Standard SGML unterscheiden zwischen Markup-Sprache (SGML selbst), Dokumenttyp-Definitionen (SGML-DTDs; dort werden die für einen Dokumenttyp zulässigen Markups oder Tags und zulässige Dokumentstrukturen ähnlich der BNF einer Programmiersprache definiert), Dokumenten (zu einer DTD gehörend) und Formatierungsstilen (DSSSL definiert). Die Trennung erlaubt unterschiedliche Formatierung von Dokumenten für verschiedene Ausgabegeräte und -Formate (als A4- oder A5-Dokument, für Bild-

schirmfenster mit Rollbalken und separatem Inhaltsverzeichnis-Fenster, usw.). In der Tat ist jede HTML-Version eine SGML-DTD; WWW-Browser sind aber keine SGML-Parser, so daß für jede neue HTML-Version eine neue Browser-Version entwickelt werden muß, die genau diese SGML-DTD verarbeiten kann. Formatierungsstile gibt es nicht, stattdessen formatiert der Browser nach fest codierten Algorithmen.

Derzeit wird versucht, die genannten Nachteile zu beseitigen, ohne die hohe Komplexität von SGML ins Internet zu übernehmen: als abgespeckte Version von SGML soll XML an die Stelle von HTML treten, ergänzt durch XSL zur Definition von Formatierungsstilen (ersetzt zaghafte HTML-basierte Ansätze wie „*cascading style sheets*“, CSS). WWW-Browser werden als XML-Parser komplexer, aber hoffentlich langlebiger. DTDs sind nicht zwingend vorgeschrieben.

**WWW als Hypermediasystem.** Betrachtet man WWW als Hypermedia-System, so bringt die HTML-Lastigkeit weitere Nachteile. HTML verleitet dazu, große Dokumente statt semantisch „atomarer“ Knoten zu verfassen. Ein HTML-Dokument wird damit quasi zu einem Metaknoten, Absätze oder Unterkapitel sind Knoten vergleichbar. Damit aber entsteht ein Bruch zwischen Hypertexten als vernetzten HTML-Dokumenten und Sub-Hypertexten in HTML-Dokumenten. Bei letzteren dominiert die hierarchische logische Dokumentstruktur (z.B. Kapitel, Unterkapitel, Paragraph, Aufzählungselement), die als Baum mit Querverweisen interpretiert werden kann. Das Dexter-Referenzmodell wird nicht realisiert, da Links (Universal Resource Locators, URLs, in WWW-Terminologie) in Textdokumente eingebettet sind anstatt in einer getrennten Schicht zu residieren; entsprechend können Links nicht rückwärts traversiert werden (welche URLs verweisen auf HTML-Dokument X?) und Konsistenz ist nicht gewährleistet (URLs veralten oder werden ungültig). Quellanker sind nur für Text und wenige weitere Medien definiert („*Imagemap*“-Grafiken, VRML-Szenen), die meisten Medien werden dadurch zu „*Sackgassen*“ der Hypertext-Navigation. Zielanker sind nur rudimentär implementiert, meist führen URLs an den Anfang des bezeichneten Zielknotens. Knotentypen werden nur auf einer systeminternen niedrigen Stufe unterscheiden, nämlich als Medientypen, benutzerdefinierte Knoten- und Linktypen dagegen nicht unterstützt.

Einige dieser Nachteile könnten mit XML und verwandten Bestrebungen beseitigt werden. So erlaubt XML die Knotenbeschreibung mittels strukturierter Metadaten, Knotentypen lassen sich über DTDs definieren. Xlink ermöglicht Linktypen, rückwärts-traversierbare und n-äre Links sowie Systemarchitekturen nach dem Dexter-Modell.

**WWW als globale Datenbank.** Mittels RDF können ausschnittsweise Netz- (d.h. Graph-) Strukturen beschrieben werden. Zusammen mit XML können so komplexe Hypertext-Netze konsistent erzeugt und besser indiziert werden. Die Leistung von Suchmaschinen kann verbessert werden. Erste Forschungsansätze lassen auch hoffen, daß Lesen im Abfrage-Modus (siehe Abschnitt 2.5.1) durch Hypertext-Anfragen in Zukunft breit unterstützt wird.

**WWW als Plattform für verteilte Programme.** Java-Applets sind ein Beispiel für dynamische (Code speichernde) Hypertext-Knoten. Deren Integration mit Information speichernden Knoten wird im sog. „*Document Object Model, DOM*“ geregelt: dort wird festgelegt, wie Programme auf Elemente eines Markup-Dokuments zugreifen. Als nächster Integrationsschritt ist zu erwarten, daß für viele Anwendungsbereiche DTDs entstehen, die auch von Programmen genutzt werden: XML könnte zur Internet-weiten abstrakten Syntaxnotation werden, in der zusammengesetzte Datentypen (Überweisungsformulare) einheitlich beschrieben werden. Java-Erweiterungen für Prozedurfernaufruf und Internet-Agenten könnten WWW zur vollwertigen Programmierplattform ausweiten (dazu und zu konkurrierenden Ansätzen siehe Kapitel D10.5.2).

## Allgemeine Literatur

- Koegel Buford, J. F.: Multimedia systems. Reading, Mass.: Addison-Wesley 1994  
 Steinmetz, R.: Multimedia-Technologie: Grundlagen, Komponenten und Systeme. 2. Aufl. Berlin: Springer-Verlag 1999  
 Steinmetz, R.; Nahrstedt, K.: Multimedia: Computing, communications and applications. Englewood Cliffs: Prentice-Hall 1995

## Spezielle Literatur

- [CACM 94] Communications of the ACM (special issue on hypermedia) 37 (1994) 2  
 [Fluckiger 95] Fluckiger, F.: Understanding networked multimedia: applications and technology. London: Prentice-Hall 1995  
 [Gibbs 95] Gibbs, S. J.; Tsichritzis, D. C.: Multimedia programming: objects, environments and frameworks. New York: ACM Press 1995  
 [Heinrichs 96] Heinrichs, B.: Multimedia im Netz. Berlin: Springer-Verlag 1996  
 [IMA 95] Interactive multimedia association: multimedia systems services. Draft Recomm. Practice 1995  
 [Multimedia Systems 95] Multimedia systems. 3 (1995) 1, Berlin: Springer-Verlag 1995  
 [Nielsen 95] Nielsen, J.: Multimedia and hypertext: the internet and beyond. Cambridge, MA: AP Professional 1995  
 [Reimers 97] Reimers, U. (Hrsg.): Digitale Fernsehtechnik: Datenkompression und Übertragung für DVB. 2. Aufl. Berlin: Springer-Verlag 1997  
 [Steinmetz 95] Steinmetz, R.: Analyzing the multimedia operating system. IEEE MultiMedia 2/1 (1995) 68-83  
 [Taylor 89] Taylor, M. M.; Néel, F.; Bouwhuis, D. G. (eds.): The structure of multimodal dialogue. Amsterdam: Elsevier Science Publishers B.V. 1989  
 [W3C] <http://w3c.org>: Internet-Adresse des World-Wide-Web Consortiums mit Verweisen auf aktuelle Versionen der Dokumente zu WWW-Standards

## 3 Datenbanksysteme

K. Dittrich

3.1	Anforderungen und Eigenschaften	878
	Datenbasis, Datenbank – Dauerhaftigkeit – Integrität – Anwendungsunabhängigkeit – Flexibilität und Bequemlichkeit – Größe – Mehrfachbenutzbarkeit – Integration	
3.2	Datenmodelle und Datendefinition	879
3.2.1	Konzepte	880
3.2.2	Das Relationenmodell	882
	Struktur – Konsistenzregeln – Datenbanksprache	
3.2.3	Weitere konventionelle Datenmodelle	886
3.2.4	Datenmodelle mit Objektkonzepten	886
	Objektorientierte Datenmodelle – Objektrelationale Datenmodelle	
3.3	Datenzugriff und -manipulation	889
3.3.1	Konzepte	890
3.3.2	Anfragesprachen	891
	Daten auffinden – Daten ändern – Daten löschen – Daten einfügen – SQL und Programmiersprachen – Anfragesprachen für Datenmodelle mit Objektkonzepten	
3.3.3	Sichten	894
3.3.4	Standards	896
3.4	Steuerung und Überwachung	896
3.4.1	Zugriffsschutz	896
3.4.2	Konsistenzsicherung	898
3.4.3	Transaktionen	900
3.4.4	Verteilung	903
	Verteilungsentwurf – Ausführung verteilter Anfragen und Manipulationsoperationen – Bearbeitung verteilter Transaktionen	
3.4.5	Speicherverwaltung und Datentransport	905
3.5	Datenbankentwurf	906
3.6	Analytische Datenbanken	907
	Online Analytical Processing (OLAP) – Data Mining – Data Warehouses	
3.7	Datenbanken und semistrukturierte Daten	909
3.8	DBMS als Komponenten von Informationssystemarchitekturen	910
3.9	Weiterentwicklung	910
	Aktive Datenbanksysteme – Temporale Datenbanken – Parallele Datenbanksysteme – Datenbanksysteme im Internet – Multimedia-Datenbanksysteme – Echtzeitdatenbanksysteme	
	Allgemeine Literatur	911
	Spezielle Literatur	912

Bei fast allen Automatisierungsaufgaben der Informatik ist es erforderlich, daß Programme Daten inspizieren oder bearbeiten können, die irgendwann in der Vergangenheit (meist von anderen Programmen) produziert worden sind, und daß umgekehrt von ihnen selbst erarbeitete oder aus der Umwelt aufgenommene Daten für künftige Verwendungen