Referat:

Graphische Benutzeroberflächen und die Multimediafähigkeit

Multimediale Standards von ODA bis MPEG

zum Seminar: Hypermedia im computerunterstützten Lernen

von Metin Çetinkaya und Frank Gadegast

gehalten am 23. Juni 1993

Addressen: **Metin Çetinkaya**
Allerstraße 10
12049 Berlin
GERMANY

Fon: ++ 49 30 6223102

E-mail: brain@cs.tu-berlin.de

Frank Gadegast
Leibnizstraße 30
10625 Berlin
GERMANY

Fon/Fax: ++ 49 30 3128103

E-mail: phade@cs.tu-berlin.de
phade@contrib.de

Inhaltsverzeichnis:

I. Einleitung

1. Motivation

"Es hieße Eulen nach Athen tragen, in der Welt offener Systeme ein weiteres Mal den Sinn von Standards zu begründen" **[KRÜ92]**.

Dies soll hier auch nicht versucht werden. Jedoch ist es auch in der sich rasch vergrößernden Gemeinde von Multimedia-Standards, wie z.B. der "**S**tructured **G**eneralized **M**arkup **L**anguage" (SGML), "**Hy**permedia/**Time**-based Document Structering Language" (HyTime) oder **JPEG** und **MPEG** notwendig zu zeigen, wie und daß diese Standards bereits in bestehende Systeme integriert und neue Applikationen mit auch für den Standard-Benutzer nützlichen Funktionen geschaffen wurden.

"People share common languages because they need to communicate with one another. A common structuring language for hypermedia documents is needed in order to permit human communication in this relatively new combination of media, given that the computing environments we create for ourselves are now and will probably always be heterogeneous." **[NEW91]**

Es soll ein Hauptschwerpunkt diese Referats sein, diese Integrationen und Neuerungen dieser multimedialen Standards und damit verbundene neue Begrifflichkeiten vorzustellen und einzuordnen, Vor- und Nachteile dieser Systeme und Standards aufzuzeigen und deren möglichen Einsatz im Bereich des computerunterstützten Lernens zu zeigen.

2. Definitionen

Der Begriff des Dokuments wird gegenwärtig definiert als "eine Menge von Informationen, die zur menschlichen Wahrnehmung bestimmt ist" **[BOR91]**.

Multimedia-Dokumente sind Dokumenten mit audio- oder visuellen Erweiterungen oder anderen multimedialen Verknüpfungen.

Der Begriff des "**I**ntegrated **O**pen **H**ypermedia" **[GOL90]** (IOH) beschreibt die drei Grundpfeiler der offenen Dokumentbearbeitung. Standards in diesem Bereich beschreiben Dokumente, die eben:

- *integrierte*: jede Information ist verknüpfbar,
- *offene*: die Adressierung der Daten ist systemunabhängig
- *Hypermedia-*: multimediale, nicht unbedingt sequentiell aufgebaute Dokumente

Dokumente sind.

II. Offene Dokumentverarbeitung mit multimedialen Standards

1. ODA

Der "**O**ffice **D**ocument **A**rchitecture" (ODA) - Standard wurde 1986 als "ISO 8613: Information Processing - Text and Office Systems - Office Document Architecture and Interchange Format" verabschiedet **[ISO87]**.

In dem Dokumentmodell von ODA (mittlerweile wird ODA jedoch eher als "**O**pen **D**ocument **A**rchitecture" übersetzt) wird die Inhaltsarchitektur eines Dokumentes in zwei grundlegende Bereich getrennt:

- der logischen Struktur (z.B. die Gliederung der Kapitel)
- der Layout-Struktur (z.B. der Seitengestaltung, wie sieht die Kapitelüberschrift aus)

Deshalb werden in ODA Dokumentklassen und deren Regelwerk beschrieben. Logische Strukturen dieses Regelwerks sind z.B. Begriffe wie "Kapitel", "Absatz" oder "Fußnote", diese sind jedoch nicht im ODA-Modell selbst verankert. Layout-Objekte sind z.B. ein Inhaltsverzeichnis, ein Rahmen oder ein Block. Die Baumstruktur der sog. "Basic Objects" ergeben zusammen die ein ODA-Dokument.

Es ergibt sich folgender Editierprozess:

- **Editing process:** The document editing process is concerned with creating a new document or modifying a previous one. Upon completion of editing, the resulting document can be interchanged. Such a document is said to be interchanged in "processable" form; it is suitable for input to either the editing or layout process.

- **Layout process:** The document layout process is concerned with defining a page-oriented organization (i.e. a layout) for the document content. The layout process can generate a document which is not intended to be modified, the "formatted" form; it is suitable only for input to the imaging process.

- **Exchange process:** The last process can also generate a "formatted processable" form which can be processed further if desired; it is suitable for input to any of the imaging, layout or editing process **[HOE89]**.

Weiterhin soll ODA mit seinen standardisierten Dokumentaustauschformaten einen weitgehenden offenen Dokumentaustausch ermöglichen.

1.1 ODIF

Normalerweise sollten ODA-Dokumente mit Hilfe einer aus der "**A**bstract **S**yntax **N**otation **1**" (ASN.1) (wurde als ISO 8824/25 1990 verabschiedet **[ISO89]**) abgeleiteten Kodierung repräsentiert werden. Diese Art der Repräsentation von ODA-Dokumenten wird "**O**ffice **D**ocument **I**nterchange **F**ormat" (ODIF) genannt **[ISO90]**.

Da die Implemetierungen von SGML (siehe unten) jedoch weit fortgeschrittener sind, als die von ODA, werden ODA-Dokumente auch in SGML kodiert. Diese Art der Repräsentation von ODA-Dokumenten nennt man ODL.

1.2 ODL

ODL ist zwar ein von SDIF verschiedener Standard, benutzt jedoch das gleiche Kodierungsschema (siehe deshalb 2.1.5).

1.3 HyperODA

HyperODA ist der (noch in der Entwicklung befindliche) Standard, der ODA um die benötigten Hypermedia-Eigenschaften erweitert. Er besteht (momentan) aus vier Teilen, die die folgende Schwerpunkte behandeln:

- Referenz-Links
- Zerteilen / Zusammenführen von ODA-Dokumenten

- Zeitrelevante Erweiterungen und
- Hyperlinks: d.h. Links innerhalb und zu anderen ODA-Dokumenten.

1.4 Resümee

Bereits bei der Verabschiedung von ODA 1989 ist die Basisfunktionalität des ODA-Standards als sein einziges größeres Manko angesehen worden. Bereits damals hat man das Fehlen der folgende Funkionalitäten erkannt:

- Farbe
- Sicherheitsaspekte
- Komplexe Tabellen
- Mathematische Formeln
- Integration von Daten oder Bürografik
- Serienbrieffunktionen
- Inhaltsverzeichnisse oder Indexlisten
- nicht-rechteckige Layout-Objekte

Auch das Fehlen von multimedialen Objekten (z.B. Audio, Video, Links, etc.) ist nicht zu übersehen. Es ist zu befürchten, daß ODA durch die notwendige und nicht zu verhindernde Standardisierungsflut (und die enorme Komplexität einer ODA-Implementierung) nicht den Weg zur weitverbreiteten Anwendung findet.

Standards, die nicht bis zur Standard-Benutzer-Ebene in der Softwareentwicklung vordringen und somit nicht Einzug in die tägliche Softwarenutzung finden, werden von der Industrie meist nicht weiter unterstützt; nicht umsonst gib es mehr Industrie- als Internationale Standards. Je komplexer internationale Standards werden, desto mehr applikationsabhängige Formate wird es geben.

2. SGML

Die "**S**tructured **G**eneralized **M**arkup **L**anguage" (SGML) wurde für das 'Anwendungsszenario Verlagswesen' entwickelt und 1986 als ISO-Standard 8879 verabschiedet. Sie ist eine (Meta)Sprache zur Auszeichnung von Dokumentstrukturen. Mit ihr kann eine logische Auszeichnung der Dokumente vorgenommen werden, in dem Dokumentstrukturen mit einem "deskriptiven Markup" versehen werden. Deskriptiv; die SGML-Notation eines Dokuments ist also menschenlesbar.

Die Auszeichnungsinformationen eines Dokuments werden von den Inhaltsinformationen mit Hilfe von Zeichen abgegrennt, die im normalen Text nicht vorkommen dürfen. Die Auszeichnungsinformation plus die Abgrenzungszeichen (z.B. <PARA>) nennt man "tag".

Da SGML nur die generelle Notation zur Verfügung stellt, wurden weiterhin mehrere SGML-Industrie-Standards vereinbart, wie z.B. der AAP-Standard (**A**ssociation of **A**merican **P**ublishers), die DFN (Deutsche Forschungsnetz) - Dokumenttypen (DAPHNE) oder die "Struktext"-Definition des Bundesverbands Druck. Diese weitergehenden Festlegungen nennt man "Document-Type-Definition" (DTD) oder Styles.

"The architecture of an SGML document is expressed in its Document Type Definition. The syntax is expressed as a set of elements each with its own generic identifier (i.e. a tag), an optional set of attributes and attribute data types, and a 'content model' a [...] production stating what sort of data [...] can be placed inside each element." **[NEW91]**

"Styles are used to combine processing parameters (of "tags") into manageable sets. Styles may be activated by linking them to a logical categorization." **[BOR91.2]**

2.1 SDIF

Mit ISO 9069 wird auch ein allgemeines, anwendbares Austauschformat für SGML-Dokumente festgelegt: das "**S**GML **D**ocument **I**nterchange **F**ormat" (SDIF) **[ISO90.2]**.

"Nach Angabe des zugrundeliegenden Zeichensatzes folgt die Übertragung des eigentlichen Dokumentinhalts in der zeichenorientierten SGML-Notation. Abschließend können noch ausgelagerte *Entity*-Definitionen folgen. Diese

verschiedenen Komponenten des Austauschformats werden mit Hilfe eines ASN.1-Rahmens identifiziert und damit gegeneinander abgegrenzt." **[BOR91.3]**

Gerade diese ausgelagerten Entity-Definitionen ermöglichen bereits dem SGML-Standard in der jetzigen Form auch multimediale Objekte systemunabhängig zu kodieren, wie z.B. Telex-Zeichen, Rasterbilder oder die Geometrie-Grafik des "**G**raphischen **K**ern **S**ystems" (GKS). Diese werden wie gewohnt (oder in bereits bestehenden Standards definiert) binär am Ende eines SGML-Dokuments gespeichert. Innerhalb des Dokuments wird mit einem *Entity*-Tag auf den externer Datenblock verwiesen.

2.2 Resümee

Mit diesem Standard waren zwar auch keine multimedialen Objekte beschreibbar, jedoch alle anderen der bereits existierenden, dokument-verarbeitenden Applikationen konnten bereits Gebrauch von SGML machen.

Da insbesondere die Funktionalität z.B. der Texverarbeitungen oder "**D**esk**t**op-**P**ublishing" (DTP) - Programme sich immer mehr anglich und angleicht ist der Fortschritt in diesem Bereich in einer systemunabhängigen Datenverarbeitung, d.h. im Dokumentaustausch, zu suchen. Auch im Sinne der immens steigenden (gespeicherten) Informationsflut ist dieser Schritt notwendig.

Der Rest dieses Referats soll sich mit solchen existierenden und geplanten Applikationen und Integrationen beschäftigen.

3. HyTime

Was SGML zur Kodierung von multimedialen Objekten fehlt, sind:

- Referenzierung unterschiedlichst festgelegter oder benannter Teile eines Dokuments
- systemunabhängige Auflösung von Objektpfaden und -applikationen
- die Möglichkeit der Beschreibung der bivalenten Eigenschaften von multimedialen Objekten (z.B. Video: wird, wenn integriert in einem Dokument vorhanden, als Standbild angezeigt, kann jedoch auch 'abgespielt' werden)
- Event- und Zeitsteuerung der Objekte

Die "**Hy**permedia/**Time**-based Document Structering Language" (HyTime) wird auch die Hypermedia-Erweiterung von SGML genannt, ist jedoch (oder wird es werden) ein eigener Standard: ISO 10744 **[ISO90.3]**. HyTime wurde zuerst für die "**S**tructured **M**usic **D**escription **L**anguage" (SMDL) **[ISO90.4]** entwickelt, später jedoch, als man sah, daß dessen Möglichkeiten auch für andere Bereiche genutzt werden konnten, abgespalten.

HyTime bietet zur Abdeckung möglicher Adressierungsprobleme verschiedener Möglichkeiten, um dokumentinterne und -externe 'Fundorte' von multimedialen Daten zu bezeichnen. Diese sind grob unterteilbar in Namensräume, Koordinatensysteme und Eigenschafts- und Zeiträume.

Da nicht alle Objekte in einem Hypermedia-Dokument durch einen exakten Namen zu referenzieren sind, bietet HyTime die Möglichkeit der Koordinatenreferenzierung (wie aus der Geometrie bekannt), der baumartigen Referenzierung oder der semantischen Bezeichnung.

Vor allem aber ist auch die Zeit und das Tempo spezifizierbar, und zwar nicht nur in festgelegten Zeiteinheiten, sondern auch semantisch und in Bezug auf andere Objekte oder Ereignisse.

Einer der wichtigen Gründe für die Modularität des Entwurfs liegt darin begründet, daß nicht jedes Dokumente sämtliche Hypermedia-Funktionen besitz und benutzt. Daraus folgt das HyTime genauso zur Kodierung von einfachen (d.h. nicht zusammengesetzten) Hypermedia-Objekten benutzt werden kann, wie z.B. SGML zur Kodierung von Plain-ASCII-Text. Es geht hier also nicht nur um Hypermedia-Dokumente, sondern auch vorrangig um Austauschformate.

"HyTime is designed to help application developers to avoid duplication of effort, and to minimize the difficulty and complexity of making their applications conform to the standard." **[NEW91]**

Die Abstraktheit des Entwurfs wird im Verhältnis von HyTime zu SGML deutlich. Formal ist Hytime eine weitere DTD in SGML. Inhaltlich soll es aber gerade keine DTD sein, sondern eine 'Meta-DTD', die das Definieren eigener

Hypermedia-DTD's erlaubt. Praktisch bedeuted dies, daß "Diskussionsergebnisse über anwendbare HyTime-DTD's noch ausstehen, von realen Implementierungen einer HyTime-Engine ganz zu schweigen" **[KRÜ92]**.

3.1 Dokument-Processing in HyTime

Die HyTime-Struktur überläßt der Applikation die komplette Kontrolle betreffs der "events", der Darstellung und sowie der Änderung und zeitlichen Veränderung der Objekte innerhalb eines Dokuments.

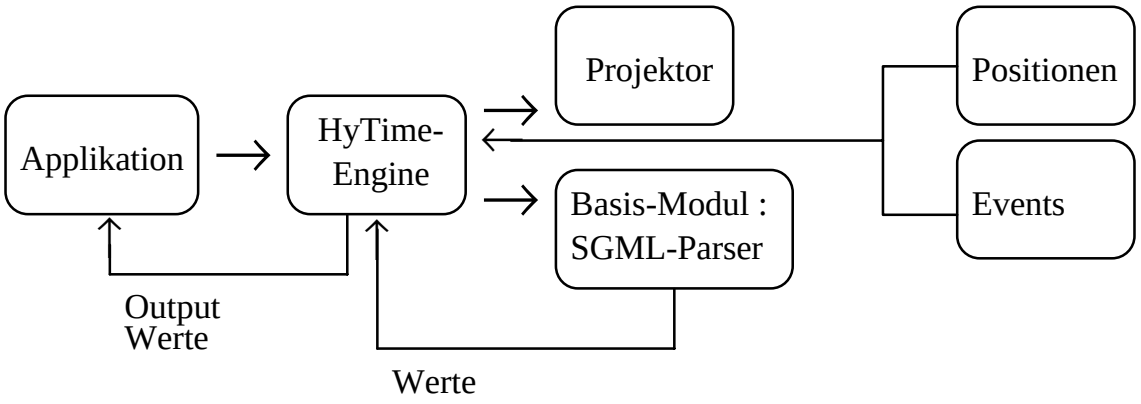


Bild I:

HyTime.

Will eine Applikation ein HyTime-Dokument bearbeiten, ruft es die notwendige Funktionen der HyTime-Engine auf. Diese wiederum wertet das (Teil-)Dokument mit Hilfe des im Basis-Moduls bereitgestellten SGML-Parsers aus. Der Parser gib alle ermittelten Werte an die HyTime-Engine zurück, diese wiederum gibt alle ermittelten Daten und den Parser-Output an die Applikation zurück. Die Applikation kann diese Daten ignorieren, da es die Aufgabe der HyTime-Engine ist, die Objekte (eben grad durch den entsprechenden, externen Projektor des Ojekttyps) des Dokuments darzustellen.

Nachdem das Dokument analysiert ist, beginnt HyTime mit der Auswertung von Positionen im FCS, überprüft "events", ruft Projektoren zur Darstellung auf, etc. . Dabei wird das Dokument (oder dessen Objekte) von Ausgangszustand (engl. "source FCS") mit Hilfe der "event-", "wand-" und "baton-"Definitionen in einen Endzustand (engl. "target FCS") projeziert.

(**Anmerkung:** HyTime ist 1992 vom ISO als International Standard verabschiedet worden, ist jedoch noch nicht in Papierform erhältlich gewesen. Dieser Abschnitt des Referates bezeiht sich deshalb auf die Dokumentation und den Draft, die bis zu diesem Zeitpunkt erhältlich war.)

4. Beispiele

4.1 ISOTEXT

Das hier am Fachbreich 20 unter der Führung von Ute und Carsten Bormann Anfang 1986 entwickelte ISOTEXT-Projekt ist einer der wenigen wirklich bekannten ODA-Editor/Formatier-Systeme. Es ist noch nicht fertiggestellt, bzw. sind noch nicht alle Komponenten der ODA/HyperODA-Architektur integriert.

ISOTEXT ist ein interaktiver WYSIWYG-Editor für ODA-Dokumente und wurde mit Hilfe von Elk, Scheme, den Programiersprachen C und C++ und Motif unter X11 auf Sun-Workstations implementiert. Das Einzigartige an diesem Projekt ist die Zerlegung des Editors in Einzelkomponenten wie es der Editier-Prozess vorgibt. Nicht-serielle Dokumente verlangen nach einer nicht-seriellen Verarbeitung der *Events* und die Verteilung der Aufgaben auf mehrer Client-Server-Applikationen.

ISOTEXT kann man getrost als "großes" System verstehen, momentan besteht es aus ca. 150.000 Zeilen C und C++ Source-Code, das resultierende ausführbare File ist ca. 9MB groß. Zusätzlich sind noch einige externe Programme zu nennen, die in Verbindung mit ISOTEXT entstanden, wie z.B. der Rasterbild-Editor XPED **[BOR91.2]**.

4.2 Andere ODA-Editoren

HERODE ist ein auf dem ECMA Standard 102, das ist der Vorgänger von ODA, basierender WYSIWYG-Editor, der ebenfalls unter X11 auf Sun Workstations läuft und in Objective C implementiert wurde. Näheres und neueres ist hierzu nicht bekannt **[KRÖ.91]**.

Phillips Papyrus ist während der Verabschiedung des ODA-Standards entstanden und war zunächst auf ECMA 101 beschränkt. Es bietet jetzt ähnliche, doch im Vergleich zu ISOTEXT reduzierte, Features. Nachteil des Papyrus-System ist vor allem die Hard- und Software-Platform auf der es aufsetzt, einem auf 386/PC-basierendem Unix-Derivat mit eigener Windows-Umgebung. Außerdem sind (durch leichte Änderungen im Datenformat von Papyrus) beide System miteinander unkompatibel.

4.3 CALS

CALS ist eine der ersten, echten SGML-Applikationen und, wen wunderts, eine Entwicklung des U.S. Department of Defense (U.S. DoD). Die schier unglaubliche Menge an technischen Manuals und Sicherheitsdaten hat das D.o.D schon früh gezwungen, nach einer Methode des Datenaustauschs zu suchen (allein die Army-interne Telefonliste verbraucht jeden Monat drei Tonnen Papier und kostet in Herstellung, Druck und Verteilung ca. 55.000 Dollar; im Manual-Archiv des D.o.D sind insg. ca. 600.000 Seiten abgelegt und gespeichert).

Das "**C**omputer-aided **A**cquisition and **L**ogistic **S**upport program" (CALS) ist ein ganze Familie von Programmen mit dem Ziel, eine papierlose Armee zu schaffen (hierzu wurden die Applikationen auf dem grundlegenden Standard "Interactive Electronic Technical Manual" (IETM) aufgebaut). Die ersten Programme dieser Familie wurden 1989 auf Basis des SGML-Standards geschaffen, um **Online-Manuals** zu ermöglichen. Mit Hilfe der SGML-Tags ist es nun weiterhin möglich, technische Manuals nicht nur als platformunabhängige Texte zu verstehen, sondern auch als technische Datenbank.

Ein weiteres Projekt von CALS ist "Electronic Review of Documents" (ERD). Dieses Projekt soll Versionskontrolle von Dokumenten bereitstellen und benötigt folgende Funktionalitäten:

- associate (hyperlink) comments with text elements at any level in the document structure.
- provide comment ID information (e.g. review, date, priority, classification/category, status).
- provide for configuration/version control of comments.
- provide structures for indentification of replacement text, in case where precise insertion data is know.
- provide structures to support sorting comments related to a particular topic of concept ("key" or "topic" attribute, user-defined values, etc.).

The [ERD-]group is pursuing the idea of basing the CALS ERD capabilities on HyTime, because of HyTime's ability to support:

- the creation of specific logical addresses within both text and graphic documents.
- the creation of hyperlinks to document addresses that will work predictably on dissimiliar platforms.
- a standard representation for representing the various kinds of activities associated with reviewing documents **[NEW91]**.

Es muß hier noch einmal betont werden, das für die geplante Erweiterung von CALS zu CALS ERD keinerlei Änderungen, weder von SGML noch von HyTime, nötig sind.

4.4 AAP und ACM

Eine Schwierigkeit bei der Entwicklung von SGML-Applikationen ist die enorme Komplexität der SGML-DTD's. "Developing a DTD is roughly as complicated as devolping a compute language - not a computer program, but a computer language" **[HAY92]**. Insbesondere die "**A**ssociation of **A**merican **P**ublishers" (AAP) und die "**A**ssociation of **C**omputing **M**achinery" (ACM) haben sich bei der Entwicklung von DTD's hevorgetan, ihre DTD's werden oft genug als Ausgangsbasis für andere DTD-Entwicklungen genutzt **[HAY92]**.

4.5 Carousel

Adobe System Inc., bekannt geworden durch die "Erfindung" von Postsript, mußte nach der Sättigung des Postscript-Marktes nach neuen Märkten suchen. Insbesondere drängten andere Postscript-Derivate bei Laserdruckern auf den Markt. Und neue Software-Entwicklungen sind in diesem Bereich kaum notwendig, fast jedes Betriebsystem und jede Oberfläche hat bereits einen Postscript-Viewer. Postscript ist eine Seitenbeschreibungssprache und somit nicht interactiv, die Entwicklung der entsprechenden Software ging schnell von statten.

Deshalb versucht Adobe die Vision des papierlosen Büros voranzutreiben. Mit Carousel verwandelt sich nun der PC in eine Zeitschrift **[TAN92]**.

Im Frühjahr 1993 sind mehrere Komponenten des Carousel-Systems für MacIntosh und PC unter der Oberfläche MS-Windows erscheinen. Der Kern des System ist ein SGML-Dokument-Driver. Dieser übersetzt die Formate der bekanntesten Text- und DTP-Programme beider Rechner-Familien in ein Carousel-internes und SGML-basiertes Datenformat. das Driver/Viewer Paket ist für nicht mehr als 200 Dollar zu haben sein. Weitere Programme werden die Konvertierung von PostScript zum Carousel-Format, die Übernahme von gescannten Seiten und Textretrieval ermöglichen.

4.6 DynaText

DynaText von Electronic Book Technologies (EBT) ist eigentlich ein Software-System für Online-Delivery von elektronischen Büchern. Um jedoch Platformunabhängigkeit und eine Separierung onv Format und Inhalt zu erreichen, wurde als grundlegendes Format SGML benutzt. Insbesondere verhält sich DynaText wie ein Hyperlink-System (multi-windowing ist möglich, Links können selber definiert werden), jedoch sind noch keine Multimedia-Erweiterungen wie HyTime integriert.

Eigene DTD's und Linktypen (z.B. im Vorgriff auf HyTime können auch externe Multimedia-Applikationen per Linkdefinition eingebunden werden) sind mit DynaText entwickelbar. Außerdem können mit Hilfe der mitgelieferten Toolkit-Library eingene User-Interface für Navigation, Darstellung und Retrieval in SGML-Dokumenten entwickelt werden.

"EBT's software is library of objects and methos contructed as a system integrator toolkit. The toolkit has facilities for data retrieval, hypertext navigation, stylesheet rendering and gereating a user interface. EBT uses its toolkit to create a shrink-wrapped product: the DynaText Browser. Users may license the standard product or create their own browser using the same toolkit as EBT. This is not as difficult as it may sound: EBT says its DynaText Browser is only 30-50 lines of code that interact with the toolkit." **[WAL92]**

Für die weitere Zukunft hat EBT die Unix-Manual-Pages im Auge. In einer Zeit, in der man 80 mal 24-Character-Bildschirme getrost als "Computer-Schrott" bezeichnen kann, sind insbesondere diese Manuals einer erweiterten Dokumentverarbeitung zuzuführen. Sie eignen sich auch besonders gut, da schon jetzt in den Manuals Struktur- und Hyperlink-Informationen enthalten sind. Sie werden jedoch nur von dem Manual-Retrieval-System "xman" der X11-Oberfläche genutzt. Editierung im WYSIWYG-Stil ist damit jedoch auch nicht möglich.

DynaText ist auch für die MS-Windows Oberfläche auf PC zu haben (genaue Fakten lagen hierzu jedoch nicht vor; es ist nicht bekannt, inwieweit z.B. die Toolkit-Library auf MS-Windows portiert wurde). Eine Mac-Version is für das nächste Jahr geplant.

4.7 Iris Insight

Iris Insight ist eine ganze Familie von Kundensupport-Produkten, incl. Support-Information, System-Diagnose, elektronischem Technik-Support und der Online-Dokumentation. Alle Produkte dieser Familie wurden von Silicon Graphics (SGI) in einer einheitlichen graphischen Benutzeroberfläche (X11 mit Motif) zusammengefaßt. Grundlage des Systems ist ein SGML-Parser. Es besteht aus den folgenden Modulen:

- **Task Assist:** Dieses Modul ordnet die Dokumentation in einer Baum-Struktur und wurde von SGI selber entwickelt. Es ist kein SGML-Viewer, eher ein Datenbank-Utility, daß die "tags" der in SGML gespeicherten Information auswertet.

♦ **Diognostics:** Dieses Modul ist bis jetzt wirklich nur ein Diagnoseprogramm. In einer späteren Version soll es System-Fehlermeldungen auf die dementsprechende Online-SGML-Dokumentation abbilden können.

♦ **Electronic services:** Dieses Modul soll technischen Support via Modem (etc.) ermöglichen und hat keinerlei Bezug zu SGML.

♦ **Document Library:** Dies ist eine Bibliothek der auf CD-Rom gelieferten technischen Referenz-Manuals für die Iris Workstation von SGI. Der SGML-Viewer "Insight" wird als Lizenz-Produkt von Electronic Book Technologies (EBT) mitgeliefert. Zusätzlich wird noch eine SGML-basierte "how-to"-Library mitgeliefert, SGI-interne Informationen, die in Buchform nicht mehr zu veröffentlichen wäre.

Der Viewer ist ein erweiterte Version des DynaText-SGML-Editors von EBT Version 1.5. Diese Erweiteungen betreffen nur Audio-, Video-, Animations- oder dreidimensoniale Graphic-Software, der Rest entstand mit Hilfe von EBT's DynaText-Toolkit-Library. SGI liefert momentan nur den Viewer, d.h. der Kunde kann nur dem System Informationen hinzufügen, wenn er selber einen SGML-Editor besitzt (z.B. die Vollversion von DynaText). Zukünftig will SGI jedoch auch die Vollversion ausliefern, um den Kunden die Möglichkeit zu geben, Markierungen oder persöhnliche Anmerkungen in der SGML-Dokumentation zu machen.

Version 2.0 von DynaText wurde im August 1992 angekündigt und beinhaltet dann auch die "TEX"-Syntax und erweiterten Graphic-Support, sowie HyTime-konforme HyperText-Links.

4.8 Avalanche

Avalanche Development Co. hat bereits mehrere Programm-Pakete im Bereich Dokumentverarbeitung und SGML produziert. Nennenswert sind in unserem Zusammenhang folgende Produkte:

♦ **FastTAG:** ermöglicht das automatisierte Verbinden und Konvertierung von elektronisch gespeicherten Textfiles sowie, gescannten Dokumenten (incl. Bildern etc.). FastTAG läuft auf den unterschiedlichsten Platform und erleichtert den Austausch zwischen bereits entwickelten Dokumenten und SGML.

♦ **SGML Hammer:** ermöglich den umgekehrten Weg; von SGML zu den eigenen Dokumenformaten. Somit können SGML-Dokumente auch weiterhin von Systemen verarbeitet werden, die noch keine SGML-Editoren haben.

♦ **Proof Positive:** Ist die SGML-Erweiterung für das Interleaf 5-Publishing-System. Somit sind Dokumente mehrer Autoren, Stil-Familien etc. möglich.

4.9 Wordperfect

Die Unix-Version von Wordperfect (WordPerfect Corp.) ist schon jetzt SGML-fähig, d.h. sie ließt nicht nur SGML-Dokumente ein, sondern ist intern auf SGML aufgebaut und unterstützt dementsprechend SGML beim Editieren und Kodieren. Im soeben im Frühjahr 1993 erschienene MS-Windows-Version von Wordperfect unterstützt ebenfalls SMGL.

Diese "kleine" Nachricht wird den Software-Markt entscheidend verändern, da somit SGML auch in Systeme der durchschnittlichen Preisklasse Einzug hält. Ausßerdem sind die jeweiligen Wordperfect-Versionen upgrade-fähig. Daraus folgt, daß der Standard-Benutzer im Zuge der normalen Weiterentwicklung eines Produktes SGML erwerben wird.

4.10 Diverse SGML/HyTime-Produkte

Diverse SGML- und SGML-verwandte Produkte sind von den folgenden Firmen bereits entwickelt (echte SGML-fähige Produkte sind mit ♦ einem, Produkte mit SGML-Schnittstellen mit einem ◊ gekennzeichnet):

- ♦ Agfa Corp, (CAPS Shared Document Managment System (SDMS))
- ♦ ArborText Inc. (ArborText ADEPT Line)
- ♦ Folio (Folio Views)
- ♦ Frame Technology Corp. (Frame Builder)
- ♦ Information Dimensions Inc. (BASISplus erlaubt speichern und darstellen von SGML-Doumenten)
- ♦ Interleaf Inc (RDM Interleaf 5, Interleaf 5 [SGML])
- ♦ Office Workstations (Guide)
- ♦ Telecommunications Industry Forum (EID)
- ♦ Xsoft, a division of Xeox Corp. (GlobalView DocuBuild)
- ♦ Manfred Krüger hat soeben eine "DTD for elctronic delivery of newspapers" veröffentlich, in ihr sind bereits rudimentäre Ansätze von HyTime enthalten. Eine HyTime-Engine ist in Arbeit.
- ♦ Das "International Commitee for Accessible Document Design" hat eine DTD für Braille- (Blinden-) Schrift vorgestellt.
- ♦ Die "Davenport Group" wird für Ende 1992 eine komplette HyTime-basierte SGML-architectur vorstellen (incl. HyTime-Engine und Applikationen).

◊ Aldus Corp. (PageMaker)
◊ Digital Equipment Corp. (Team Document Library)
◊ Quark Inc. (Quark Pubishing System QuarkXPress)
◊ Ventura Software Inc. (Ventura Publisher)

♦◊ sowie diverse Produkte von SoftQuad Inc., Diabold Inc., Electronic Data System Corp., Andersen Consulting, Computer Task Group, El Segundo, Graphic Communications Association, Bantam Doubleday Dell Publishing Group, U.S. Securities and Exchange Commission, the Pentagon etc.

(Für weitere Informationen muß hier auf Yuri Rubinsky's Artikel **[RUB92]** verwiesen werden).

5. Resümee

Zusammenfassend ist festzustellen, daß der Markt und der Nutzungsgrad der Offenen Dokumentverarbeitung in den nächsten Jahren ungeahnte Größen annehmen wird.

Aufgrund der jetzt schon verabschiedeten und von der Industrie notwendigerweise verwendeten Standards wird der sog. Publishing Software Markt (Umsatz 1991: SGML 204 Millionen Dollar, insg. 1.62 Milliarden Dollar) im Jahr 1995 einen Gesamtumsatz von ca. 3.4 Milliarden Dollar (SGML 550 Millionen Dollar) erreichen. Das entspricht einer Verdopplung in nur vier Jahren. Dies sind Steigerungsraten, wie sie seit Einführung der 5. Rechnergeneration nicht mehr erreicht wurden (und dies ist gerechnet, ohne den Softwareschub von anderen Standards wie HyTime oder ODA/HyperODA zu berücksichtigen) **[DOL92]**.

"Taking lots of information on the road in a portable form is a goal in document processing."

III. Bild-,Video- und Audio-Formate

Wir wollen hier wiederum nur die Formate betrachten, die auch als internationale Standards verabschiedet wurden. PD-Standards (wie GL, FLI, DL etc.) sollen hier nicht und Industrie-Standards (wie DVI, AVI etc.) sollen hier nur am Rande betrachtet werden.

Die digitale Dastellung eines Studio-TV-Signals erfordert gemäß der CCIR Recommendation 601 eine Nettodatenrate von 166 Mbit/s. Keine der bereitstehenden digitalen Medien kann eine solche Datenrate bieten. Mit den hier zu behandelden Standards können jedoch auch Video- und Audio-Signale auf bereits bestehende Medien übertragen und gespeichert werden.

1. JPEG

Der internationale Standard JPEG (Joint Picture Experts Group) (ISO/IEC SC29/WG10) ist ein Standard zur Codierung von photographischen Standbildern. Der Basisalgorythmus beschreibt eine Transformationskodierung auf der DCT.

1.1 YUV-Farbraum

Da sich gezeigt hat, daß das menschliche Auge Farbwerte in einer geringeren Auflösung als Helligkeitswerte wahrnimmt, wird der gängige RBG-Farbraum (Rot-Grün-Blau) bei den digitalen Bildstandards in drei Planes umgewandelt, eine Luminanz-Plane (Grauwerte, Y) und zwei Chrominanz-Planes (Farbwerte U/V, manchmal auch Cr/Cb). Die Farbwertplanes können dann von ihrer Kantenlänge halbiert werden (das ist dann ein Viertel der Fläche !). Dies reduziert die Datenmenge enorm, ist jedoch ein Datenverlust. Diese Art der Kodierung nennt man den 4:1:1 YUV-Farbraum.

1.2 Diskrete Cosinus Transformation

Aus den zur Verfügung stehenden Transformationen hat sich die DCT (Dicsrete Cosine Transform, abgeleitet aus der Diskreten Fourier Tranformation) als besonders effizient erwiesen. Eine auf einen 8x8 Pixelblock angewendete DCT ergibt wiederum einen 8x8 Pixelblock. Die Koeffizienten der DCT lassen sich als Spektrum des 8x8 Eingabeblock interpretieren.

Der Koeffizient mit den Frequenzen Null in horizontaler und vertikaler Richtung wird als DC-Koeffizient bezeichnet. Die restlichen 63 Koeffizienten werden als AC-Koeffizienten bezeichnet. "Während die Energie des Bildsignals zufällig verteilt sein kann, konzentriert sich die Energie des korrespondierenden DCT-Blocks vorzugsweise auf Koeffizienten mit niedrigen Frequenzen" **[MUS93]**. Werden die Koeffizienten im Zick-Zack durchnummeriert, ergeben sich als zu speichernde Werte ein DC-Koeffizient, dann wenige niedrige AC-Koeffizienten und viele AC-Koeffizienten nahe Null. Die DCT ist ein verlustfreies Verfahren, da die Kodierung komplett umkehrbar ist.

Nun eignen sich diese Werte jedoch optimal um sie mit dem Huffman-Verfahren (eine Art Top-Down Shannon-Fano-Kodierung; häufige Bytewerte werden durch kurze Bitfolgen ersetzt) und anschließend nach dem Lauflängen-Verfahren zu kodieren (sollten sieben Nullen im Strom nacheinander folgen, braucht man nur die 7 und die 0 zu kodieren).

Da man aber beim JPEG-Verfahren auch die Kompressionsrate angeben kann (die höherfrequenten Signale des Blocks werden dann ignoriert), fassen wir die Kodierung daher wie folgt zusammen:

- verlustbehaftete Kodierung
- 4:1:1 YUV-Farbraum
- DCT - Discrete Cosine Transform
- Entropie und Huffmann-Codierung

2. H.261

"Der H.261-Standad kann zur Kompression von Bildsequenzen für Videokonferenzen oder für Bildfernsehen verwendet werden." **[MUS93]**. Zusammenfassend hat H.261 folgende Chrakteristika:

- Standard zur Übertragung von digitalen Video-Sequenzen der CCITT
- bei p*64 kbit/s
- bereits als Hardware vorhanden
- einfache Umsetzung RGB $\Rightarrow$ YUV
- Intra-Frame-Kodierung (mit Prädiktion)
- auch DCT und Huffmann

3. MPEG-I

"Coded Representation of Picture, Audio and Multimedia/Hypermedia Information"

MPEG heißt "**M**otion **P**icture **E**xpert **G**roup" und ist die Gruppe der ISO, die sich mit der Standardisierung im Video-Bereich beschäftigt. Im Dezember 1992 wurde ein Draft International Standard (DIS 11172) mit dem übersetzten Titel "Kodierung von Bewegtbildern und assoziierten Audio für digitale Speichermedien mit bis zu 1,5 MBit/s" vorgelegt.

3.1 MPEG-Draft

Dieser DIS (ISO/IEC JTC1/SC2/WC11) ist in drei Einzel-Standards unterteilt:

Video	- Kodierungstechniken etc. entsprechend JPEG (s.o.)
Audio	- Psychoakustisches Modell
System	- behandelt Synchronisaton und Multiplexing (auf diesen Teil wird hier durch fehlende Informationen nicht näher eingegangen)

Ein MPEG-Stream läßt sich also in mindestens 32 Video- und Audio-Spuren und 2 Systemspuren zur Synchronisation zerlegen.

3.2 MPEG-Begriffe

Ein MPEG-Video-Stream wird durch Aneinanderreihung von Intra- (I), Predicted- (P), und Bidirectional- (B) beschrieben. I-Frames sind komplett im JPEG-Format abgelegt, in P-Frames werden nur die Differenzen zu einem I-Frame kodiert, ein B-Frame kodiert die mittleren Differenzen eines I- und eines P-Frames. Jeder Frame wird wiederum in drei Planes, eine Luminanz-Plane (Grauwerte, Y) und zwei Chrominanz-Planes (Farbwerte, Cr und Cb), zerlegt. Alle Planes werden in sogenannte Macroblöcke unterteilt. Alle Planes werden in 8x8 Pixel-Blöcke aufgeteilt. Jeder dieser Blöcke wird mittels einer Discrete-Cosine-Transform (DCT) kodiert, dabei wird der erste Wert jedes Blocks als DC-Koeffizient (DC) bezeichnet; die restlichen Differenzwerte als AC-Koeffizienten (AC). DC's werden untereinander auch als Differenzen gespeichert. "Echte" DC's bezeichne ich als Master-DC's (MDC); einen als Differenz zu einem MDC gespeicherten DC als Differenz-DC (DDC).

Zusätzlich können Frames auch als Verweise der Macroblöcke auf gleiche, in vorherigen Frames gespeicherten Blöcken kodiert werden (hier bezieht man sich jedoch of 16x16 Pixel-Blöcke). Dieses Verfahren nennt man Motion Compensation.

3.3 MPEG-Parameter

Im Gegensatz zu JPEG werden in MPEG feste Quantisier- und Huffmann-Tabellen verwendet, diese sind zwar nicht für jede Video-Übertragung optimal, können dann jedoch auch in Hardware kodiert werden. Dies ist unerläßlich für die Fernsehtechnik.

Die Bildgröße ist variabel (in 16-Pixel-Schritten) und kann maximal horizontal 720 pels und vertikal 576 pels betragen, dies ist die Auflösung eines normalen S-VHS Signals.

Bei einem zu erzielenden Datendurchsatz von 1,5 Mbit/s werden maximal 386 Kbit/s für den Audio-Bereich verwandt.

Durch das Kodierungverfahren, sind Einzelbildzugriff, schnelles suchen (Positionierung auf I-Frames), rückwärts spielen und die Editierbarkeit des Datenstroms gegeben.

MPEG wurde durch die Wahl der Kodierung auf asymetrische Anwendungen (Fernseh-Technik, Multimedia-Mail, etc.) zugeschnitten, mit vermehrtem Hardware-Einsatz können jedoch auch symetrische Verfahren implementiert werden. Insbesondere benötigt die Dekodierung keine Hardwareunterstützung.

3.4. Resümee

MPEG ist nicht die optimalste Videokodierung, allerdings ist sie der einzig internationale Konsens in diesem Bereich und wird, auch dank seiner Verwandschaft mit den Telekommunikations-Standards (H.261, S-VHS) weite Verbreitung finden.

4. MPEG-Audio

Der Audio-Teil des MPEG-Draft beschreibt Mechanismen und Algorithmen mit denen die digitale Speicherung von Audiosignalen auf kostengünstigen Speichermedien auf der einen Seite und die digitale Übertragung von Audiosignalen auf Kanälen mit begrenzter Kapazität auf der anderen Seite ermöglicht werden sollen.

4.1. Technische Daten

Studioformat

Die Darstellung eines stereophonen Audiosignals im Studioformat erfordert eine Abtastfrequenz von 48kHz und eine gleichförmige Quantisierung von 16bit pro Abtastwert. Daraus ergibt sich eine Datenrate von 768kbit/s für ein Monosignal, als Produkt der Multiplikation der 48kHz mit den 16bit/Abtastwert. Daraus resultierend ergibt sich für ein Stereosignal eine Datenrate von 2x768kbit/s, also ca. 1,5Mbit/s.

Als Vergleich dazu wird auf einer Compact Disk mit einer Abtastfrequenz von 44,1kHz bei der gleichen Quantisierung von 16bit/Abtastwert gearbeitet, somit ergibt sich für ein stereophones Signal eine Datenrate von 2x706kbit/s, also ca. 1,4Mbit/s.

MPEG-Audio-Standard

Im MPEG-Audio-Standard werden zwei Abtastfrequenzen verwendet, nämlich zum einen 32,441kHz und zum anderen 48kHz. Aber im Gegensatz zu den oben beschriebenen Fällen ergeben sich hier im Endeffekt Datenraten zwischen 32kbit/s und 192kbit/s für ein Monosignal. Für ein Stereosignal ergeben sich Datenraten zwischen 128kbit/s und 384kbit/s.

4.2. Ziele

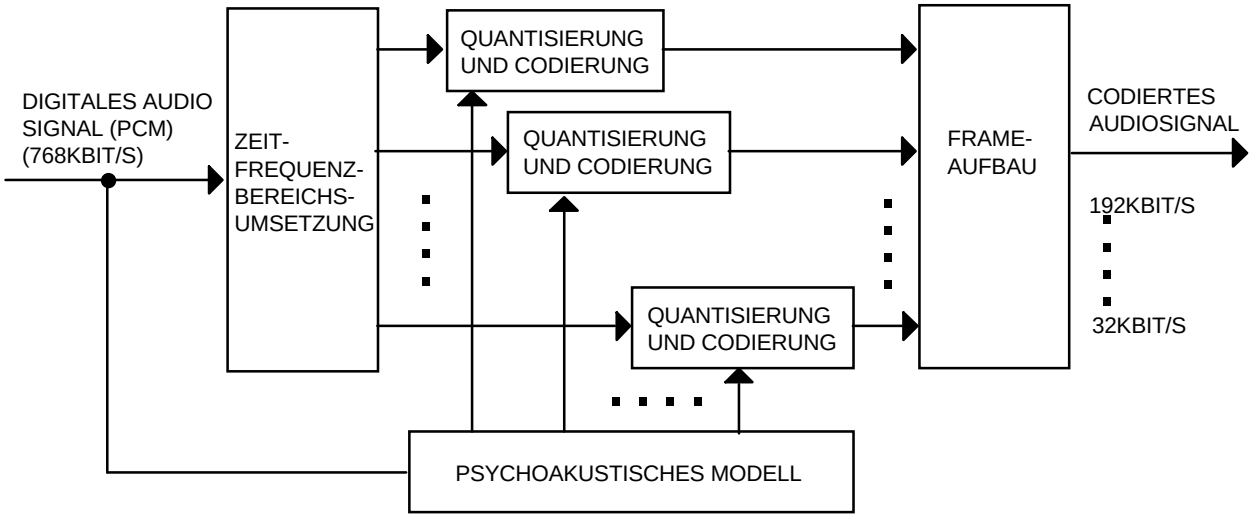
Das Ziel des Standards ist mit einer von 1,5Mbit/s im Studioformat auf 256kbit/s reduzierten Datenrate eine der Qualität einer Compact Disc ebenbürtige Qualität zu erreichen, wobei auch bei niedrigeren Datenraten wie 192kbit/s bis hinunter zu 128kbit/s noch akzeptable Qualitäten erzielt werden. Hierbei ist jedoch die Einschränkung zu machen, daß in diesen unteren Datenraten-Bereichen bei kritischen Testsignalen Qualitätsminderungen durch das menschliche Gehör wahrnehmbar sind. Das menschliche Gehör ist im allgemeinen ja sowieso bei Störungen im Audio-Bereich empfindsamer als im visuellen Bereich, daher wird eine der CD vergleichbare Qualität angestrebt.

4.3. Codierungsmodi

Innerhalb der Codierung sind verschiedene Modi zu unterscheiden, genauer gesagt sind es vier an der Zahl. Als erstes steht das *Single Channel Coding zur* Verfügung, das zur Codierung von Monosignalen herangezogen wird, dazu kommt noch das *Dual Channel Coding* als Mittel zur Codierung von z.B. bilingualen Monosignalen (wie z.B. Zweikanalton im Bereich des TV, wo auf jedem der zwei Kanäle eine andere Sprache ausgestrahlt wird). Weiterhin existiert das *Stereo Coding* zur Codierung eines Stereosignals, bei diesem Verfahren werden die beiden Kanäle trotzdem separat codiert. Als letztes ist das *Joint Stereo Coding* zu nennen, das ebenso wie das *Stereo Coding* zur Codierung eines Stereosignals benutzt werden kann, jedoch mit dem einen Unterschied, daß bei diesem Verfahren die Datenredundanz und -irrelevanz zwischen den beiden Kanälen ausgenutzt und somit eine Datenverminderung erreicht werden soll. Dieses zuletzt genannte Verfahren ist jedoch noch nicht realisiert.

4.4. Grundlegendes Codierungskonzept

Das digitale Eingangssignal wird in 32 gleichförmige Spektralkomponenten (Frequenzgruppen, Teilbänder) zerlegt, dieses Grundprinzip entspricht dem Vorgang im menschlichen Gehör (Psychoakustik). Der Vorgang wird als Zeit-Frequenzbereichs-Umsetzung bezeichnet. Die Spektralkomponenten werden dann in Abstimmung auf die Wahrnehmungseigenschaften des menschlichen Gehörs codiert. Diese Codierung wird von einem der drei Layer durchgeführt. Die Quantisierung und Codierung wird unter Einbeziehung einer Maskierungsschwelle realisiert. Diese Maskierungsschwelle wird vom *Psychoakustischen Modell* für jede Komponente individuell durch eine sogenannte *Discrete Fourier Transform* berechnet und gibt die maximal erlaubte Quantisierungsfehlerleistung an, mit der noch codiert werden darf, ohne daß eine Wahrnehmung dieses Fehlers durch das menschliche Gehör befürchtet werden muß. Es ist ein sehr sensibles Verfahren, da bei dem kleinsten Fehler, sprich Abweichung von der Maskierungsschwelle, eine Störung durch das Gehör wahrnehmbar würde.
Nach der Codierung der einzelnen Spektralkomponenten wird der Frame zusammengesetzt und als codiertes digitales Audiosignal ausgegeben, das sich zwischen 32kbit/s und 192kbit/s bewegt.



4.5. Die drei Layer

Die oben erwähnten drei Layer des MPEG-Audio-Standard arbeiten alle nach dem beschriebenen Grundprinzip, jedoch gibt es zwischen ihnen natürlich Unterschiede, die im folgenden angeschnitten werden sollen.

Layer I

Es werden Frames zu je 384 PCM-Abtastwerten verarbeitet, was bei einer Abtastfrequenz von 48kHz einem Zeitintervall von 8ms entspricht. Bei einer Zerlegung in 32 Spektralkomponenten oder auch Teilbänder ergibt das 12 Abtastwerte pro Teilband, die Teilbandabtastwerte. Aus diesen wird der maximale Absolutwert bestimmt und mit 6 Bit kodiert, dieser

Wert wird Skalenfaktor genannt. Zuletzt wird der Wert unter Einbeziehung der Maskierungsschwelle quantisiert und codiert.

In diesem Layer wird mit einer Datenrate von 384kbit/s eine vergleichbare CD-Qualität erreicht.

Layer II

Im Gegensatz zum Layer I werden hier Frames zu je 1152 PCM-Abtastwerten verarbeitet, was bei einer Abtastfrequenz von 48kHz einem Zeitintervall von 24ms entspricht. Das restliche Verfahren ist dem des Layer I entsprechend, außer daß sich durch die Verdreifachung der Framelänge 36 statt 12 Teilbandabtastwerte ergeben. Durch die entsprechende Quantisierung und Codierung wird in diesem Layer mit einer Datenrate von 256kbit/s eine vergleichbare CD-Qualität erreicht.

Dieser Layer ist im Verhältnis zum Layer I komplexer aber auch effizienter.

Layer III

Der Layer III besitzt die größte Komplexität und zugleich die höchste Effizienz. Die Anzahl der Abtastwerte und Zerlegung in Spektralkomponenten entspricht Layer II, ergänzend werden im Layer III die Abtastwerte am Ausgang eines jeden der 32 Teilbänder einer MDCT (Modified Discrete Cosine Transform) zugeführt, die dann 18 Spektralkoeffizienten pro Teilbandsignal erzeugt. Hierbei kann je nach Anwendung die spektrale Auflösung der MDCT für ausgewählte oder alle Teilbänder auf 6 Spektralkoeffizienten reduziert werden. Dieses Verfahren nennt sich adaptive Fensterumschaltung und sorgt für eine höhere zeitliche Auflösung bei Wahl einer geringeren Frequenzauflösung (sprich 6 Spektralkoeffizienten pro Teilband statt 18). Hieraus kann man folgern, daß eine geringere Frequenzauflösung eine höhere zeitliche Auflösung zur Folge hat und umgekehrt.

Auf die quantisierten Werte wird zuerst eine Huffman Codierung und zuletzt eine Lauflängencodierung vorgenommen. Ansonsten entspricht das Verfahren in diesem Layer dem des Layer II.

Die CD-Qualität wird hierbei mit der selben Datenrate wie in Layer II erreicht, bei niedrigeren Datenraten erreicht man noch akzeptable Ergebnisse, mit den oben bereits erwähnten Einschränkungen.

4.6. Resümee

Die in Layer I und Layer II beschriebenen und verwendeten Coder sind bereits fast vollständig als integrierte Schaltungen realisiert und werden wohl bald zur Verfügung stehen, falls sie denn nicht schon zur Verfügung stehen, da die Layer I Codierung bereits in DCC-Recordern (Digital-Compact-Casette-Recorder) und einigen Multimedia-Systemen verwendet.

Die Layer II Codierung wird vom europäischen Digital Audio Broadcasting (DAB) System verwendet werden.

Zukünftiges Ziel dürfte es sein mit dem Layer III die CD-Qualität schon bei einer Datenrate von 2x64kbit/s zu erreichen, hierzu wird eine ideale Realisierung des *Joint Stereo Coding* notwendig sein. Dies ist in naher Zukunft zu erwarten.

Aufgrund dieser Entwicklungen in diesem Bereich darf man im Hinblick auf Multimedia-/Hypermedia-Systeme auch Fortschritte erwarten, da durch die Reduzierung der Datenmengen die Einbindung von Video- und Audiodaten einen realistischeren Bezug erhält.

5. Sicherheitmechanismen für Multimedia-Daten (Integrität)

5.1. Fehlerarten und Fehlerpropagierung

Je nachdem wo in dem MPEG-Video-Stream Fehler bei der Übertragung bzw. Speicherung auftreten, sind auch die zu beobachtenden Fehler des Videobildes unterschiedlich. Da es sich bei MPEG um ein hochkomprimierendes Datenformat (fast redundanzfrei) handelt, sind Fortpflanzungsfehler (Fehlerpropagierung) nicht zu vermeiden und bedürfen einer besonderen Analyse.

5.1.1 Fehler innnerhalb von Macro-Blöcken

Sollte ein AC-Koeffizient fehlerhaft sein, hat dies zur Folge, daß der Rest des Blockes auch fehlerhaft wird. Da jedoch AC's untereinander als Differenzen gespeichert werden, ergibt sich ab dem fehlerhaften Wert eine Helligkeitsverschiebung. Ist der Fehler zwischen Original-Wert und Fehler-Wert nicht zu gross (< 32, d.h. ein Bitkipper in den unteren 5 bits eines bytes), kann diese Verschiebung toleriert werden, dies wären 5/8 aller Bitkipper, sofern sie nur diesen Makroblock (siehe unten) betreffen. Allerdings machen AC's nur ca. 25-40 % der Videodaten aus.

Sollte ein DDC-Koeffizient fehlerhaft sein, ist nicht nur der gesamte Macroblock (im oberen Sinne) betroffen, sondern auch auch alle nachfolgenden Blöcke, deren DC's als Differenz zu diesem DDC gespeichert werden. Dies ist im günstigsten Falle kein weiterer Block. Im MPEG-Stream ist jedoch nur ca. jeder 60ste Macroblocke mit einem MDC gespeichert, d.h. das im ungünstigsten Fall ein fehlerhafter DDC ca 60 weitere Blöcke beeinflußt.

Diese Beeinflußung geschieht in der oben beschriebenen Verschiebung im Helligkeitsbereich.

Bild II:

Fehlerhafter Grauwert-Macroblock mit folgender Helligkeitsverschiebung des restlichen Frames.

Sollte ein MDC-Koeffizient fehlerhaft sein, beeinflußt er nicht nur den kodierten Macroblock, sondern ALLE nachfolgenden, mit einem DDC-Koeffizienten kodierten Blöcke (ca. 60). Diese Fehlerart kann nur tolerierbar sein, wenn es sich um einen fortlaufenden und nur einmal verwendeten Video-Stream handelt.

5.1.2 Unterschiede von Fehlern in Y- und Cr-/Cb-Planes

Da das menschliche Auge unempfindlicher für Farben als für Grauwerte ist, wurden die Chrominanz-Planes bei der MPEG-Kodierung halbiert. Fehler innerhalb von Macroblöcke (wie oben skizziert), sollten nun größere Auswirkungen auf Farb-Macroblöcke haben als auf Grauwert-Macroblöcke, da die Farb-Macroblöcke größer sind (16x16 Pixel).

Dies trifft in der Praxis um so eher zu, da Fehler in den Chrominanz-Planes eher zu einer totalen Zerstörung der folgende Farb-Werte führen (bis zum nächsten MDC-kodierten Macroblock). Dies kann größere Ausmaße annehmen, und wird vom Auge eher wahrgenommen, als eine gleichgroßflächige Helligkeitverschiebung.

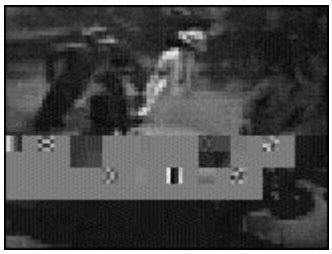


Bild III:

Fehlerhafter Farbwert-Macroblock mit folgender Zerstörung der nächsten Macroblöcke. Dieser Frame "repariert" sich durch den nächsten MDC-kodierten Macroblock selber .

Besonders sind Farbverschiebungen in Computeranimationen als sehr störend anzusehen, denn hier werden Flächen eher durch die Farbgebung als durch unterschiedlich Grauwerte voneinander abgegrenzt. Ausserdem werden zumeist nur begrenzt viele Farben des Farbraums benutzt. Eine Verschiebung in großen gleichfarbigen Flächen ist als sehr störend zu empfinden. Solche Animationen zeichnen sich auch durch eine verstärkte Motion Compensation (siehe unten) im Farbbereich aus, was wiederum zu einer verstärkten Fehlerpropagierung führt.

Fehler in den Chrominanz-Planes von Computer-Animationen sind nicht tolerierbar.

5.1.3 Fehler in P- und B-Frames, Motion Vektoren oder Headern

Bisher hatten wir nur Fehler in Intra-Frames betrachtet. Da P-Frames als Differenz zu I-Frames (B-Frames sogar als mittlere Differenz von I- und P-Frames) gespeichert werden, sind diese extrem anfällig für Fehler in "ihren" I-Frames (bzw. B-Frames auch "ihrer" P-Frames). Deshalb sollte die (relative, doch dazu später) Integrität von I-Frames immer gegeben sein.

Durch Motion-Compensation werden Blöcke nicht als Blöcke sondern als Vektoren zu bereits kodierten (und gleichen) Blöcken gespeichert. Motion-Compensation bedeutet somit auch immer Fehlerpropagierung. Fehler in den Referenzframes ziehen hier räumliche Verschiebungen des Inhalt des mit Motion-Compensation kodierten Frames nach sich.

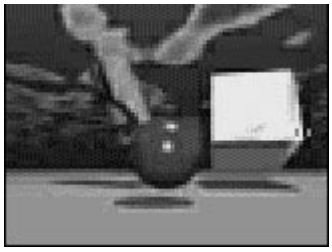


Bild IV+V:

Motion - Compensation - Fehler.
Referenzframe.
Inhaltsverschiebung.

Fehler im Header (Vorspann) sind im MPEG-Format nur zu Verzeihen, wenn es sich um einen verbindungsorientierten MPEG-Stream (Videokonferenz etc.) handelt. Ansonsten kann ein Fehler im Header (z.B. der Framegröße) den ganzen Stream zerstören. Der zusätzliche Rechenaufwand zum Schutz des Header hölt sich in Grenzen; die durchschnittliche Länge des Header ist kleiner 1 K.

Bisher hatten wir nur Fehler auf bit- bzw. Byte-Ebene (Bitkipper) betrachtet. Diese sind, wie gesehen, in bestimmten Bereichen tolerierbar. Sind jedoch ganze Bytefolgen, Planes oder Frames beschädigt, ist dies nur zu tolerieren, wenn es sich um einen fortlaufenden, einmalig zu "sehenden" MPEG-Stream handelt.

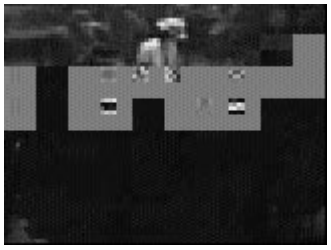


Bild VI:

Kombinierter Fehler. Der ganze Frame ist zerstört.

Lücken (meistens durch Übertragung entstanden) im MPEG-Video-Stream führen zu noch nicht genau untersuchten Fehlern. Fehlt z.B. ein sogenannter 'sequence ending code' verweigern die bekannten MPEG-Viewer die Interpretation des MPEG-Streams. Leider konnte bisjetzt noch kein lückenhafter MPEG-Stream konstruiert werden, dessen betroffener Frame auch anzeigbar wäre.

Die (jetzige) Schlußfolgerung ist, daß ein Integritäts-Check bezogen auf die Länge des Video-Streams bei Speicherung unabdingbar wird.

5.2. Szenarien

Es können verschiedene Szenarien bei der Behandlung von Fehlern (oder deren Vermeidung) in Bezug auf die Integrität eines MPEG-Stream unterschieden werden.

- **Keine Fehlerkontrolle**. Wie oben gezeigt sind hochkomprimierte Multimedia-Daten anfällig für Fehlerpropagierung. Keine Fehlerkontrolle ist nur ratsam, wenn es sich um einen "einmaligen" MPEG-Stream handelt, z.B bei Bildtelefonie oder Video-Konferenzschaltungen (ohne Aufzeichnung). Gefahrlos sind hier Fehler im Header des MPEG-Streams, da diese hier meist nicht benutzt werden.

  Eine Fehlerkontrolle bei der Speicherung durch das jeweilige Betriebsystem ist nur begrenzt gegeben. Sollte die eingesetzte Hardware jedoch auch eine leichte Fehlerkontrolle zulassen (dies sollte nicht "zu" zeitaufwendig sein), ist diese zu bevorzugen.

- **Leichte Fehlerkontrolle**. Sollen MEPG-Video-Daten für längere Zeit gespeichert oder über fehleranfällige Leitungen übertragen werden, empfiehlt sich eine teilweise Fehlerkontrolle. Die zu schützenden Teilinformation wurden durch die Fehlerbeschreibung (oben) skizziert und werden in den Lösungsansätzen weiter erläutert. Der Header sollte auch hier schon geschützt werden.

- **Totale Fehlerkontrolle**. Sollen MPEG-Video-Daten archiviert oder in professioneller Art und Weise bearbeitet (geschnitten, editiert, Tricktechnik, etc) muss eine Fehlerkontrolle auf den gesamten MPEG-Stream angewandt werden. Besonders bei der Weiterverarbeitung von Einzelbildern sind auch noch so kleine Fehler nicht zu tolerienen. Zur Fehlerkontrolle können hier jedoch bereits bekannte Verfahren eingesetzt werden (CRC-Check).

5.2.1 Leichte Fehlerkontrolle - Zusammenfassung

Folgende Grundsätze können betreffs der Integrität von MPEG festgehalten werden, wenn ein leichte Fehlerkontrolle eingesetzt werden soll:

- Es ist notwendig den MPEG-Stream zu analysieren. Eine zufällige teilweise Kontrolle des Streams bietet keinen Schutz, denn es werden nicht nur einfache Inhalte zerstört, sondern deren Folgefehler führen mit hoher Wahrscheinlichkeit zur Zerstörung des gesamten Streams. Insbesondere sind die bekannten MPEG-Player nicht gegen verkürzte MPEG-Streams geschützt; größere Systemfehler sind die Folge.

- Um nach der Analyse des MPEG-Streams nicht ein weiteres Mal Huffman dekodieren zu müssen, sollte der Original MPEG-Stream mit eingestreuten Sicherheits- oder Schutz-Informationen übertragen bzw. gespeichert werden.

Dies bedeutet nun, daß ein Sicherheitsalgorythmus folgenden allgemeinen Ablauf haben sollte:

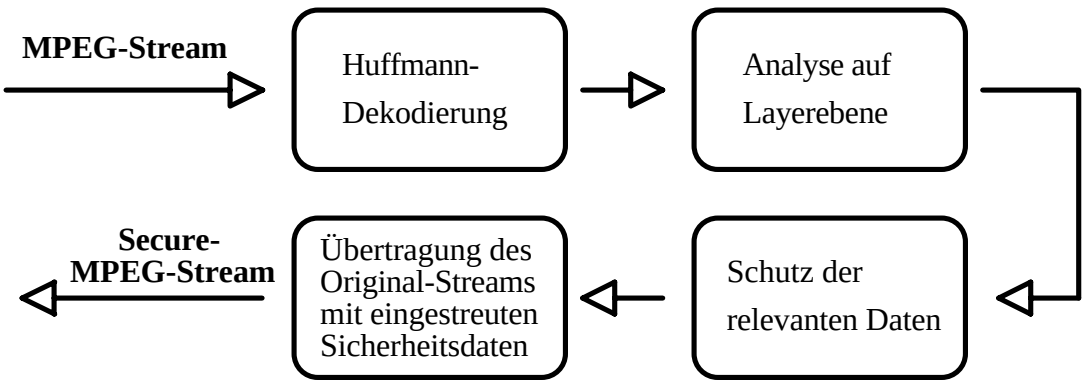


Bild VII:

Secure-MPEG-Stream-Kodierung.

Zusammenfassend sind folgende Daten des MPEG-Streams zu schützen:

- der gesamte Header
- alle DC's (MDC's sowie DDC's) beider Farb-Planes aller I-Frames
- alle MDC's der Grauwert-Plane aller I-Frames, sowie alle anderen MDC's in B- oder P-Frames, egal ob Grauwert-Plane oder Farb-Plane
- alle Motion-Vektoren
- der gesamte Trailer (falls vorhanden)

- Zusätzlich ist eine Check-Summe betreffs der Anzahl der Bytes jedes Frames zu erstellen, um eine Verkürzung des MPEG-Streams überwachen zu können. Verkürzte Frames sind beim dekodieren zu entfernen.

Es ist noch fraglich inwieweit besondere Steuerinformationen zu schützen sind. Sollten z.B fehlende 'sequence ending codes' zum Versagen der Player-Software (sowie weiterer Systemfehler) führen, ist das Vorhandensein dieser Steuerinformationen zu gewährleisten.

5.3 Resümee

Mit der beschriebenen Mechanismen wird eine Methode zur Verfügung gestellt, die bei minimalen Rechen- und Kodierungsaufwand maximalen Schutz (in Beziehung auf die Feststellung einer Veränderung der Daten) gewährleistet. Eine Fehlerkorrektur kann in Hinsicht auf das Volumen nicht zur Verfügung gestellt werden.

Verschieden Stufen des Integritätsschutzes sind denkbar. Schutz vor Veränderung der relevanten Daten in Bezug auf Manipulation, Schutz vor Unbrauchbarkeit (Versagen der Software !) oder Schutz der Abspielqualität können durch Auswahl der oben beschriebenen schutzrelevanten Daten lassen sich implementieren.

6. MPEG-II

MPEG-II wird die Erweiterung von MPEG-I werden, die Rücksicht auf die hardwaretechnischen Neuerungen und Erfordernisse nimmt. Besonders ist hierbei die neu hinzugefügte Funktionalität der "Scalability" zu nennen, die ähnlich dem Standard ODA, ein der Hardware angepasstes Konsumieren von Videos ermöglichen wird. "The ability of a decoder to ignore some portions of a total bitstream and produce useful audio and video output from the portion which is decoded." **[CHI92]**.

Weitere 'Features' von MPEG-II werden sein:

- angepaßte Bandbreite < 3 MBit/s wegen Bedarf von
  - Satellite Broadcast
  - Electronic Cinema
  - Digital Home Television, etc.

- Scalability of Audio and Video
- MPEG-1 and H.261 backward compatibility
- Multi-Channel-Mode, multi-lingual

- Encryption, Security
- MPEG-4 ? - Bitrates up to tens of kbit/s - fraktale Kompression

7. DVI

"Digital Video Interactive (DVI) ist ein früher Versuch der Firma Intel (in Zusammenarbeit mit der Firma IBM) einen Standard in der Kompression von kontinuierlichen Medien durchzusetzen. Intel kaufte 1988 die Rechte an DVI vom David Sarnoff Research Center. Ein Jahr später stellte man das erste Produkt vor. Es war ein 386er PC mit sieben Steckkarten." **[BAD93]**.

Zwei Kompressionsformate werden von DVI unterstützt: RTV (**R**eal **T**ime **V**ideo) wurde vom Benutzer mittels seiner lokalen Hardware gebraucht; PLV (**P**roduction **L**evel **V**ideo) erreichte Kompressionsraten von 160:1, konnte jedoch nur mittels von Intel authorisierten Betrieben benutzt werden.

Bis jetzt ist DVI weder auf anderen Rechnerwelten noch ohne extreme Hardwareunterstützung denkbar. Außerdem scheint sich Intel mit dem neuen Verfahren "Indeo" mittlerweile gegen das eigene DVI gestellt zu haben.

8. AVI

AVI (Audio-Video-Interlaced) scheint Microsofts Antwort auf Apple's Quicktime zu sein, kann dies jedoch nicht leisten. AVI stellt unter der Benutzeroberfläche MS-Windows 3.1 mehrere Utilities und Driver zu Verfügung, die es erlauben, kleinste Video-Sequenzen (20 Sekunden 160x120 Pixel bei 8 bit Farbe nehmen immerhin 2 MB Daten ein) per Hardware zu digitalisieren, zu editieren, komplett in die Windows-Umgebung zu integrieren, mit von Hardware gesampeltem Audio zu synchonisieren und natürlich auch alles zusammen abzuspielen.

Bild VIII:

AVI-Editor von Microsoft's Video for Windows.

Insbesondere wird hier ein Kompressionverfahren der Firma Intel (Intel Indeo) eingesetzt, das ein 2-dimensionales 'scaling' des Video-Filmes erlaubt, jedoch ist mit diesem 'scaling' nicht 'scalibility' gemeint.

Das große Manko von AVI ist die noch ungenügende Kompression, da AVI bis jetzt auch nur auf einer Hardware-Platform implementiert ist (und AVI kein internationaler Standard ist) wird sich AVI zwar einer großen Verbreitung in der Intel/Microsoft-Welt erfreuen, jedoch wohl kaum in den Bereich der Telekommunikation vordringen können.

9. Beispiele

9.1 Xing

Die MPEG-Player der Firma Xing Technologies waren die ersten PD-Programme die MPEG zu jedermann brachten. Die Software zeichnet sich durch ein sehr gute (Windows-typische) 'Usibility' aus. Xing hat auch die ersten Versuche zur Audio-Integration unternommen (Synchonisation über die Zeit mit Windows-WAV-Files).

Bild IX:

Xing's MPEG-Player-Interface.

Allerdings setzt Xing immer noch nicht auf dem MPEG-I-Standard auf. Es werden nur mit ausschließlich I-Frames kodierte Videos und keine variablen Framegrößen unterstützt.

9.2 Berkeley und Stanford

Die Gruppe um Lawrence A. Rowe, Ketan Patel und Brian Smith an der Computer Science Division-EECS, Univ. of Calif. at Berkeley stellte den ersten PD-Player auf Unix-basis vor. Dieser Code, der den kompletten MPEG-I-Standard implementiert war so einfach und 'sauber' programmiert, daß er bis heute auf dutzende verschiedene Platformen und System portiert wurde.

Ebenso das erste PD-Codec von Andy C. Hung, University of Stanford. Es erlaubt das parametrisierte Erzeugen von MPEG-Videos. Beide Tools kann man mittlerweile als den MPEG-Standard-Code bezeichnen.

9.3 NVR

NVR (**N**orth **V**aley **R**esearch, Inc.) haben mit ihrem "Digital Media Development Kit, Version 1.0" das erste integrierte, komplett interaktiv zu steuernde MPEG-System vorgestellt. Es ist für Sun-Computer unter der Benutzeroberflüche Open-Look (bez. X11) mit Hilfe von Motif erstellt worden und unterstützt den kompletten MPEG-Standard incl. 'frame-grabbing', Kodierung, Player etc.. Als kommerzielle Lösung sind jedoch die preislichen (bzw. Hardware-) Voraussetzungen (Sun IPC mit Parallax-Board plus Software = DM 20.000.-) noch als zu hoch anzusehen.

Anhang A: Quellen

[BAD93] Badura. Rolf-Stefan, Meyer-Zajontz, Jörg: Quicktime, DVI und MPEG, Referat, Jan. 93

[BOR91] Bormann, U. u. C.: Offene Bearbeitung multime. Dokumente, Informatik-Spektrum 14/1991

[BOR91.2] Bormann, C.: Open Document Processing and the ISOTEXT System, 1991

[BOR91.3] Bormann, Ute: ISIS 1 Internationale Standards for Informationstechnik - Systemarchitektur, 1991

[CHI92] Chiariglione, Leonardo: Multimedia Communication (MPEG-II), Brussels 1992

[DOL92] GDoler, Kathleen: SGML Standard Lets Firms Shift Voluminous Manuals To Computers,, Investor's Business Daily, Nov. 1992

[GOL90] Goldfarb, C.F.: The SGML Handbook, 1990

[GAD93] Gadegast, Frank: Offene Dokumentverarbeitung mit nultimedialen Standards, Referat, Jan. 1993

[GAD93.2] Gadegast, Frank, Jürgen Meyer: Integrität von Multimedialen Daten am Beispiel MPEG, Semesterarbeit, Jun. 1993

[HOE89] Hoepner, Petra: Synchronizing the Presentation of Multimedia Object, BERMMD 1989

[HUN93] Hung, Andy C.: PVRG-MPEG-CODEC 1.1, Manual, Stanford, 1993

[HUN93.2] Hung, Andy C.: PVRG-JPEG-CODEC 1.1, Manual, Stanford, 1993

[ISO87] ISO/IEC International Standard 8613/2.2: Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format, International Organization for Standardization, Geneva, 1987

[ISO89] ISO/IEC International Standard 8824/25: Information technology - Open Systems Interchange connection - Specification of Abstract Systax Notation One (ASN.1), International Organization for Standardization, Geneva, 1989

[ISO90] ISO/IEC International Standard 8824: Office Document Interchange Format (ODIF), International Organization for Standardization, Geneva, 1990

[ISO90.2] ISO/IEC International Standard 9069: SGML Document Interchange Format (SDIF), International Organization for Standardization, Geneva, 1990

[ISO90.3] ISO/IEC Draft International Standard (DIS) 10744: Hypermedia/Time-based Document Structering Language (HyTime), International Organization for Standardization, Geneva, 1990

[ISO90.4] ISO/IEC International Standard : Standard Music Description Language (SMDL), Hypermedia/Time-base Subset (HyTime), International Organization for Standardization, Geneva, 1990

[ISO92] ISO/IEC Draft International Standard (DIS) 11172: Information technology - Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbit/s (MPEG), International Organization for Standardization, Geneva, 1992

[KRÖ91] Krönert, G: Wird ODA/ODIF Bürosysteme verändern ?, 1991

[KRÜ92] Krüger, Manfred: HyTime, ix 4/1992

[LOC92] Locke, Christopher: Avalanche Development and Electronic Book Technologies Help Silicon Graphics to Deliver Documentation in SGML, Silicon Graphics World, Sep. 1992

[LOC92.2] Locke, Christopher: Foundations for Document and Information Managment, DATAPRO, McGraw-Hill, 1992

[LOC93] Locke, Christopher: CALS and SGML - Beyond Compliance and Conversion, CALS Journal, Spring 1993

[MUS93] Musman, Hans-Georg; Werner, Oliver; Fuchs, Hendrik: Kompressionsalgorythmen für interaktive Multimedia-Systeme, IT+TI 2/93, Hannover 1993

[NEW91] Stephen R. Newcomb, Neill A. Kipp, Victoria T. Newcomb: "HyTime" - The Hypermedia/Time-based Document Structering Language, Communication of the ACM, Nov. 1991, Vol. 34, No. 11

[NVR93] North Valey Reserach: Digital Media Development Kit, Reference Manual, Version 1.0, Jan. 1993

[PAT92] Patel, Ketan; Smith, Brian C.; Rowe, Lawrence A.: Performance of a Software MPEG Video Decoder, Berkeley, 1992

[RIC92] Ricciuty, Mike: Publishing Software Gets Serious, DATAMATION, Dec. 1992

[RUB92] Rubinsky, Yuri (yuri@sq.sq.com): The SGML year in review, 1992

[TAN92] Tanaka, Jamie: Does Adobe have a paper cutter?, Business Week, 16. Nov. 1992

[WAL92] Walter, Mark: SGI, Novell Pick SGML Documents, Seybold-Report on Publishing Systems, Seybold Publication, Volume 22, Number 1, 7. Sep. 1992